

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'Hamed BOUGARA – Boumerdes



Institute of Electrical and Electronic Engineering
Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of
the Requirements for the Degree of

MASTER

In Telecommunication

Option: Telecommunications

Title:

**Simulation of an IoT-based Smart Home
using Cisco Packet Tracer 7.3.0**

Presented by:

- **AROSSI Dhiya Eddine**
- **OUNASSI Omar**

Supervisor:

Dr. BELAIDI Hadjira

Registration Number:...../2020

ACKNOWLEDGMENT

*First of all, we thank **Allah Almighty** who gave us the strength and the courage to achieve this modest work.*

We would like to thank our project supervisor Dr. BELAIDI Hadjira of the Institute of Electrical and Electronic Engineering/University M'Hamed BOUGARA of Boumerdes. The door to Mrs. BELAIDI office was always open whenever we ran into a trouble spot or had a question about our research or writing. She consistently allowed this thesis to be our own work, but steered us in the right direction whenever she thought we needed it.

We also thank the members of the jury for agreeing to judge and to evaluate our work and for sharing their valuable remarks with us.

We must express our very profound gratitude to our parents, families, teachers, and friends for providing us with unfailing support and continuous encouragement throughout our years of study and through the process of researching and writing this report. This accomplishment would not have been possible without them.

DEDICATION

I would like to dedicate this project to my beloved parents, my source of inspiration and the fuel that keeps me going. To my brother Abdou, to my two sisters Zeineb and Ishraq who offered unlimited support and care, to the big family.

I would also like to thank my childhood friends whom I consider as brothers, my INELEC friends who made my university life a lot more enjoyable. To the people who care about my well-being and success.

Finally, I would like to thank my partner; Omar and everyone who contributed to the achievement of this work, and the ones who were there for me since the very beginning... Thank you all.

Dhiya Eddine

To my dear parents who have always supported me and oriented me towards the good, for their help to follow my studies, their encouragement, their prayers and sacrifices, without forgetting my sisters and brother to whom I wish a good success in their studies and work.

To my family, to my partner family. To all students of INELEC. To all those who helped me throughout my university life. Without forgetting my dear friends.

To all those who are dear to me... Thank you all.

Omar

ABSTRACT

The IoT-based Smart Home system is an integrated system built to facilitate life for people in general and especially for elderly and disabled ones; with an easy-to-use home automation system that can be fully operated based on IoT applications. The system is efficient and constructed in an easy to install, configure, run and maintain way.

This project demonstrates a system that can be applied in many areas including home security, lighting control, flame detection, smart heating, motion sensor and door control, etc... to provide its homeowner's comfort, security, energy efficiency (low operating costs) and convenience at all times. And turn on or off any appliances whenever needed to.

In this work, we have developed and implemented a smart home concept using sensors and actuators which are connected to a Gateway via Wi-Fi communication protocol. A 3G/4G client can control the home remotely using an account in an IoT server. The project is simulated using Cisco Packet Tracer simulation tool. Networking and programming are the power behind this study as it provides an interface between the sensors and the actuators and the devices to be monitored.

***Keywords:* Internet of things, Networking, Smart home, Automation, Cisco Packet Tracer.**

TABLE OF CONTENTS

Table of contents	i
List of figures	v
List of tables	viii
List of abbreviations	ix
General introduction.....	1

Chapter 1: Introduction to networks and internet of things (IoT)

1.1. History	3
1.2. Definition of a network.....	4
1.3. The objective of using networks.....	5
1.4. Classification of networks (By Dimension).....	5
1.4.1. Personal Area Network	6
1.4.2. Local Area Network.....	6
1.4.3. Metropolitan Area Network	7
1.4.4. Wide Area Network	7
1.5. Network Architecture (Model)	8
1.5.1. OSI Network Model.....	8
1.5.2. TCP/IP Network Model	10
1.6. Machine identification	11
1.6.1. MAC address.....	11
1.6.2. IP Addressing.....	11
1.6.3. The different Classes of IP addresses.....	11
1.6.4. Subnet mask	12
1.6.5. DHCP protocol.....	12
1.6.6. DNS protocol	12
1.7. Internet of things (IoT)	13

1.8. Characteristics of IoT.....	13
1.9. IoT Devices.....	13
1.10. Applications of IoT.....	14
1.10.1. The Smart Home	14
1.10.2. Smart Cities.....	14
1.10.3. Transport/Logistics	14
1.10.4. E-Health	15
1.10.5. Smart retail.....	15
1.10.6. Smart farming.....	15
1.11. Conclusion	15

Chapter 2: Smart home equipment and interconnection

2.1. Network Equipment.....	16
2.1.1. Network hub.....	17
2.1.2. Network switch	17
2.1.3. Network Router	18
2.1.4. Network modem.....	18
2.1.5. Server	19
2.1.6. Cloud.....	20
2.1.7. Cell Tower.....	21
2.2. Network Media	21
2.2.1. Twisted-Pair Cable.....	22
2.2.2. Coaxial	22
2.2.3. Fiber-optic	22
2.2.4. Wireless communication.....	24
2.3. Network Connectors	25
2.3.1. Connector definition.....	25
2.3.2. RJ connectors	25
2.3.3. BNC (Bayonet Neill–Concelman) connector.....	25
2.3.4. SC and ST connectors	25
2.4. Smart Home Required Equipment.....	25
2.4.1. Garage	25
2.4.2. Door.....	26
2.4.3. The Room.....	26

2.4.4. The Garden.....	27
2.4.5. Kitchen.....	27
2.5. Conclusion.....	28

Chapter 3: The configuration of LANs, WAN, and the IoT devices

3.1. Cisco Packet Tracer Overview.....	30
3.2. Network Layout.....	31
3.3. IPv4 addresses of the system.....	31
3.4. Internet configuration.....	32
3.4.1. Internet provider.....	32
3.4.2. Service provider.....	32
3.4.3. 3G/4G provider network.....	35
3.4.4. Home.....	37
3.5. Configuring the IoT devices.....	39
3.5.1. Wireless interface.....	39
3.5.2. Connecting devices to the Internet.....	40
3.5.3. Registration server.....	40
3.6. Interaction between Devices.....	42
3.6.1. Garage.....	42
3.6.2. Door.....	44
3.6.3. Room.....	45
3.6.4. Garden.....	47
3.6.5. Kitchen.....	50
3.7. Conclusion.....	52

Chapter 4: Tests and results

4.1. Testing the manual usage of the IoT devices.....	53
4.1.1. The garage.....	55
4.1.2. The door.....	55
4.1.3. The windows.....	56
4.1.4. The room FAN.....	56
4.1.5. The AC.....	57
4.1.6. The furnace.....	57

4.1.7. The fire sprinklers	58
4.1.8. The lawn sprinklers	58
4.1.9. The Blowers:	58
4.1.10. The siren.....	59
4.1.11. Coffee maker.....	59
4.2. Environment variables	60
4.3. Testing the automation of the IoT devices.....	61
4.3.1. The garage.....	61
4.3.2. The door	63
4.3.3. The automatic CO monitoring	65
4.3.4. The automatic temperature monitoring.....	67
4.3.5. The automatic smoke monitoring.....	70
4.3.6. The fire monitoring system	71
4.3.7. Garden water monitoring	72
4.4. Conclusion	73
General Conclusion	74
Appendix A	75
Appendix B.....	76
Appendix C.....	77
Appendix D	78
Bibliography	79
Webography	80

LIST OF FIGURES

Figure 1.1. Personal Area Network (PAN).....	6
Figure 1.2. Local Area Network (LAN).....	7
Figure 1.3. Metropolitan Area Network (MAN).....	8
Figure 1.4. Wide Area Network (WAN).....	8
Figure 1.5. OSI model layers.....	9
Figure 1.6. TCP/IP Model layers.....	10
Figure 2.1. Network Hub.....	17
Figure 2.2. Network Switch.....	17
Figure 2.3. Network Router.....	18
Figure 2.4. Cable and DSL modems	19
Figure 2.5. IBM network servers.....	20
Figure 2.6. Cloud illustration.....	20
Figure 2.7. Cell Tower.....	21
Figure 2.8. STP and UTP Cables.....	22
Figure 2.9. Coaxial Cable.....	23
Figure 2.10. Fiber Optic Cable	24
Figure 2.11. Cisco Wireless Access Point.....	24
Figure 2.12. Network connectors	25
Figure 2.13. Garage required pieces of equipment.....	26
Figure 2.14. Room required equipments	26
Figure 2.15. Garden required pieces of equipment	27
Figure 2.16. Smoke sensor	27
Figure 3.1. Cisco Packet Tracer interface	30
Figure 3.2. Our smart home network topology	31
Figure 3.3. Internet provider setting	32
Figure 3.4. DNS configuration	34
Figure 3.5. IoT service configuration	35
Figure 3.6. Backbone interface DHCP enabling	36
Figure 3.7. Cell tower interface configuration	36
Figure 3.8. 3G/4G client configuration	36
Figure 3.9. Home topology.....	37
Figure 3.10. Internet interface of the home gateway.....	38

Figure 3.11. Home gateway wireless interface configuration	38
Figure 3.12. Laptop wireless configuration.....	38
Figure 3.13. Laptop Gateway and DNS configurations	39
Figure 3.14. I/O Config TAB of an IoT device	39
Figure 3.15. wireless0 interface config TAB	40
Figure 3.16. Devices default gateway and DNS server addresses	40
Figure 3.17. IoT server field in the Config TAB of the device	41
Figure 3.18. IoT device status for the devices registered in the IoT server	41
Figure 3.19. Garage security and lighting system	42
Figure 3.20. Check the Smart LED brightness.....	43
Figure 3.21. Door's security and lighting system.....	45
Figure 3.22. Door manually monitoring.....	45
Figure 3.23. Room monitoring system design.....	46
Figure 3.24. Manual Window monitoring IoT server account.....	47
Figure 3.25. Manual Fan monitoring IoT server account.....	47
Figure 3.26. Manual Carbon Monoxide alarm monitoring IoT server account	47
Figure 3.27. garden monitoring system.....	48
Figure 3.28. Enable and disable the watering operation manually.....	49
Figure 3.29. Automatic Kitchen Fire and smoke monitoring.....	50
Figure 3.30. Coffee machine enabling	51
Figure 4.1. The IoT server interface from the 3G/4G client web browser.....	54
Figure 4.2. the authentication credential for the IoT account.....	54
Figure 4.3. Home interface of the IoT account	54
Figure 4.4. Closed garage	55
Figure 4.5. Open garage	55
Figure 4.6. Locked door	55
Figure 4.7. Unlocked door.....	55
Figure 4.8. Closed window.....	56
Figure 4.9. Open window	56
Figure 4.10. The FAN status is OFF	56
Figure 4.11. The FAN status is LOW.....	56
Figure 4.12. The FAN status is HIGH.....	56
Figure 4.13. The AC is turned OFF.....	57
Figure 4.14. The AC is turned ON	57
Figure 4.15. The furnace is turned OFF	57
Figure 4.16. The furnace is turned ON.....	57
Figure 4.17. The fire sprinkler is OFF.....	58
Figure 4.18. The fire sprinkler is ON	58
Figure 4.19. The lawn sprinkler is OFF	58
Figure 4.20. The lawn sprinkler is ON	58
Figure 4.21. The blower status is OFF	59
Figure 4.22. The blower status is LOW	59
Figure 4.23. The blower status is HIGH.....	59
Figure 4.24. The siren is OFF.....	59

Figure 4.25. The siren is ON	59
Figure 4.26. The coffee maker is OFF	59
Figure 4.27. The coffee maker is ON	60
Figure 4.28. The logical and physical view	60
Figure 4.29. Example of environmental variable setup.....	61
Figure 4.30. The change of the Sunlight (percentage) over 24 hours	62
Figure 4.31. An authorized card and motion while there is no sunlight	62
Figure 4.32. An unauthorized card and motion while there is sunlight	62
Figure 4.33. Adjusting the time to 1 AM	63
Figure 4.34. Garage security in “Waiting” state and no motion while the is no sunlight	63
Figure 4.35. An authorized card and motion while there is sunlight	64
Figure 4.36. An unauthorized card and motion while there is no sunlight	64
Figure 4.37. Adjust the time to 5:30 AM	64
Figure 4.38. Door security in “Waiting” state and no motion while there is no sunlight	65
Figure 4.39. The room window is open and the fan is working once the CO level exceeded 20% .	66
Figure 4.40. The room window is closed and the fan is OFF after the CO level is less than 2%	66
Figure 4.41. The garden windows are open and the blowers are working after the CO level exceeded 20%	67
Figure 4.42. The garden windows are closed and the blowers are OFF after the CO level is less than 2%	67
Figure 4.43. The change of the ambient temperature for 24 hours	68
Figure 4.44. The status of the furnace versus temperature variation.....	68
Figure 4.45. The status of the furnace versus the temperature decreasing.....	69
Figure 4.46. The status of the AC versus temperature decrease	69
Figure 4.47. The status of the AC with increasing temperature at several moments	70
Figure 4.48. The kitchen window state after the smoke level exceeds 20%	70
Figure 4.49. The kitchen window state after the smoke level falls under 2%	71
Figure 4.50. The siren and the kitchen sprinkler if there is no fire	71
Figure 4.51. The siren and the kitchen sprinkler in case of fire	71
Figure 4.52. The garden sprinklers if there is no fire	72
Figure 4.53. The garden sprinklers if there is fire	72
Figure 4.54. The behavior of the lawn sprinklers at three different moments	72
Figure 4.55. The behavior of the lawn sprinklers at another three different moments	73
Figure A.1. Garage lighting flowchart	75
Figure A.2. Garage lighting python code	75
Figure B.1. Room temperature monitoring flowchart.....	76
Figure B.2. Room temperature monitoring python code.....	76
Figure C.1. Kitchen smoke monitoring flowchart.....	77
Figure C.2. Room temperature monitoring python code.....	77
Figure D.1. JavaScript code for fire’s IR property	78
Figure D.2. The two states icons for the Fire	78

LIST OF TABLES

Table 1.1. The IPv4 Classes	12
Table 1.2. The default subnet masks for each class	12
Table 3.1. IPv4 addresses of all devices	31
Table 3.2. Garage security system operations	43
Table 3.3. Door security system operations	44
Table 3.4. Automatic Vent Carbon Monoxide monitoring system conditions.....	46
Table 3.5. Garden fire monitoring system conditions	48
Table 3.6. Conditions in the IoT server to automate garden watering	49
Table 3.7. Conditions in the IoT server to automate the garden CO monitoring	50
Table 3.8. Kitchen Fire and smoke monitoring conditions	51

LIST OF ABBREVIATIONS

AC	Air Conditioner
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
BGP	Border Gateway Protocol
BNC	Bayonet Neill–Concelman
CLI	Command-Line Interface
DARPA	Defense Advanced Research Projects Agency
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSL	Digital Subscriber Line
EMI	Electromagnetic Interference
HAN	Home Area Network
ICT	Information and Communications Technology
IoT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization
LAN	Local Area Network
LED	Light-emitting Diode
MAC	Media Access Control
MAN	Metropolitan Area Network
MCU	Micro-Controller Unit

NIC	Network interface card
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PAN	Personal Area Network
RFI	Radio Frequency Interference
RFID	Radio-frequency identification
RJ	Registered Jack
SBC	Single-Board Computer
SC	Subscriber Connector
SP	Service Provider
ST	Straight Tip
STP	Shielded twisted-pair
TCP	Transmission Control Protocol
TCP/IP	Transfer Control Protocol / Internet Protocol
URL	Uniform Resource Locator
UTP	Unshielded twisted-pair
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WWW	World Wide Web

GENERAL INTRODUCTION

Since the concept of home automation has become a widely interesting topic, people's expectations of how this later should be described or how the services should be provided for and accessed at home are changing frequently.

A smart home describes a residence with: appliances, lighting, air conditioning, TVs, computers, and security etc.... that can communicate with one another and that can also be controlled remotely from any room in the home; as well as, from any location in the world by phone or internet facility.

This present project consists of the design and the simulation of an IoT-based smart home using the CISCO packet tracer simulation tool. The smart home is divided into several parts inside, and can be remotely monitored by 3G/4G clients through the internet.

This project contains four chapters.

In the first chapter, a small history about networks is presented. Then we touched on defining them and we briefly explained their types. Next, we got a small explanation about the two protocol suits (models) (TCP/IP and OSI); Only the protocols that are used in our project. In the end of this chapter, a definition of the internet of things (IoT) and its characteristics and applications are mentioned.

Regarding the second chapter, loads of networks components used for establishing the internet connection and providing the home the accessibility to be controlled remotely are successfully introduced. Furthermore, the task of each component applied in each part inside are indicated.

The third chapter start by the configuration of the system using the simulation tool. First the internet is configured, then the service (DNS and IoT) provider network then the 3G/4G provider network and finally, the home network configuration is done.

In the fourth chapter, both the manual and the automatic usage of the smart home devices and the ability of accessing the home from a 3G/4G client web browser are tested.

Finally, we ended this work with a general conclusion.

1

INTRODUCTION TO NETWORKS AND INTERNET OF THINGS (IOT)

Computer Networks have become an essential tool in many aspects: human communication, gathering, exchange and sharing of information, distributed work environments, access to remote resources and many more. Starting from an historical overview, in this chapter we will define networks and describe the different types of networks and present the different protocols and architecture.

1.1. History

The foundation of computer networking was truly laid in **1958**, when researchers at Bell Labs released the Bell 101 modem. The modem was named for its function—the Modulation and Demodulation of digital information into signals that could be transmitted along a telephone wire. At the same year, **the US government founded the** Advanced Research Projects Agency (**ARPA**), which would ultimately lead to the creation of the modern Internet. ARPA's investments would become a key to the research and development of many of the protocols and systems that came to define the Internet and its predecessors, such as the Advanced Research Projects Agency Network (ARPANET).

On July 24, 1961, MIT accepted Leonard Kleinrock's doctoral dissertation, which is regarded as the first paper on "*packet switching*." This idea was a divergence from the "*circuit switching*" approach that dominated telecom at the time.

By the mid-1970s, researchers now had to figure out how to interconnect these networks. That led Kahn and Cerf to publish a Protocol for Packet Network Intercommunication, in which they first described the **Transmission Control Protocol (TCP)**. The duo continued their work on network interconnection and, in 1978, unveiled the Internet Protocol (IP), which unites the disparate networks while still allowing each one to maintain a level of autonomy within itself.

As the network of networks grew several other protocols were invented like **Domain Name System (DNS)** and routing protocols too like **The Gateway-to-Gateway Protocol, Border Gateway Protocol (BGP), Open Shortest Path First (OSPF)** which routes traffic *between* networks (not within a single network). After these protocols appeared, Berners-Lee invented the World Wide Web (www), ushering in the dot-com (.com) era, the browser wars, and eventually, the modern web as we know it today [1].

1.2. Definition of a network

Networking is referred as connecting computers electronically for the purpose of sharing information. Resources such as files, applications, printers, software and common information shared in a network. The advantage of networking can be seen clearly in terms of security, efficiency, manageability and cost effectiveness as it allows collaboration between users in a wide range. Basically, network consists of hardware component such as computer, hubs, switches, routers and other devices which form the network infrastructure. These are the devices that play an important role in data transfer from one place to another using different technology such as radio waves and wires. There are many types of network available the most common networks are Local Area Network (LAN) and Wide Area Network (WAN). While a traditional network is comprised of desktop computers, modern networks may include laptops, tablets, smartphones, televisions, gaming consoles, smart appliances, and other electronics [2].

Computer network is interconnectivity of two or more computer system for purpose of sharing data. A computer network is a communication system much like a telephone system, any connected device can use the network to send and receive information. In essence a computer network consists of two or more computers connected to each other so that they can share resources. Networking arose from the need to share resources in a timely fashion.

Sharing expensive peripherals is often promoted as the primary reason to network. But this is not a sufficient reason. In considering the cost benefits of sharing, we find some impressive arguments against networking. With today more affordable technology, we can easily dedicate inexpensive peripherals and not bother with a network. Desktops and laptops are getting less expensive as their capacities increase. As a result, the local hard disk is becoming common place and is frequently dedicated to a local desktop or laptop. Flash drives and external hard disks now have enough storage for uses.

1.3. The objective of using networks

The implementation of a local network makes it possible to offer IT services to users. Computer networks have several advantages such as:

- **Resource sharing:** It is the main objective of the computer network. The goal is to provide all the program, data and hardware available to everyone on the network without regard to the physical location of the resource and the users.
- **High reliability:** As an example of reliability, it is achieved by replicating the files on two or more machines, so in case of unavailability the other copies can be used.
- **Saving money:** This is due to the fact that the small computer has much better price to the performance ratio comparison than the large computer like mainframe, organization has preferred to install interconnected microcomputer connected to the mainframe computer.
- **Increase system performance:** Computer network have provided means to increase system performance as the work load increases (load balancing). In the days of mainframe when the system was full it was to replace with the other large mainframe computer, usually at an expensive rate not convenience for user.
- **Increases security:** Only authorized user can access resource in a computer network. Users are authenticated by their user name and password. Hence it is not possible to access the data without proper account.

1.4. Classification of networks (By Dimension)

Computer networks can be characterized by their size as well as their purpose. The size of a network can be expressed by the geographic area they occupy and number of computers that are part of the network. Networks can cover anything from a handful of devices within a single room to millions of devices spread across the entire globe [3].

1.4.1. Personal Area Network

A personal area network (PAN) is a computer network for interconnecting electronic devices centered on an individual person's workspace (see figure 1.1), typically within a range of 10 meters. However, it is possible to have multiple individuals using the same network within a residence. If this is the case, we can refer to the network as home area network (HAN) [3].

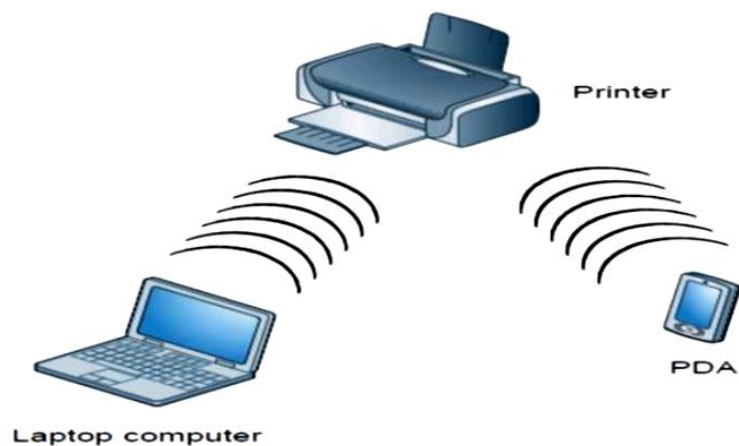


Figure 1.1. Personal Area Network (PAN) [3].

1.4.2. Local Area Network

A local area network (LAN) is usually privately owned and interconnects computers within a limited area such as a residence, school, laboratory, university campus or office building. A LAN can be as simple as two desktops and a printer in someone's home office; or it can extend throughout a company and include audio and video peripherals. Figure 1.2 is an illustration of LAN. Wireless LANs (WLAN) are the newest evolution in LAN technology [3].

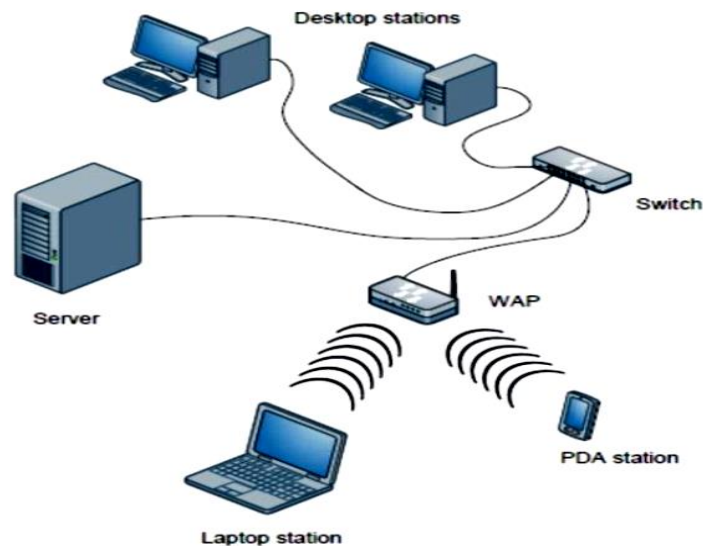


Figure 1.2. Local Area Network (LAN) [3].

1.4.3. Metropolitan Area Network

A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. The term is also used to describe the interconnection of several local area networks in a metropolitan area. High-performance routers and high-performance fiber-based connections are used, which enable a significantly higher data throughput than the internet. It is designed for customers who need a high-speed connectivity, normally to the internet, and have endpoints spread over a city or part of city [3] as shown in figure 1.3.

1.4.4. Wide Area Network

While Metropolitan Area Networks connect areas that are near each other in rural or urban areas, **Wide Area Networks (WANs)** extend across large geographic areas, such as countries or continents or even the whole world, a WAN is illustrated by figure 1.4. A WAN can be as complex as the backbones that connect the Internet or as simple as a dial-up line that connects a home computer to the internet [3].

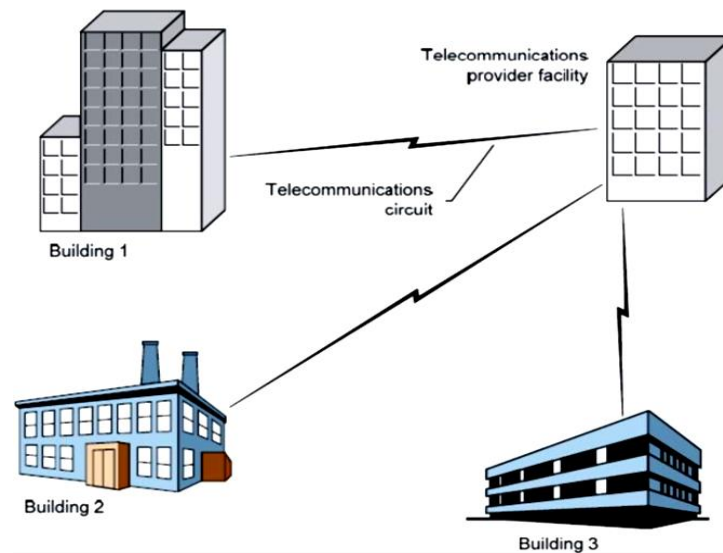


Figure 1.3. Metropolitan Area Network (MAN) [3].

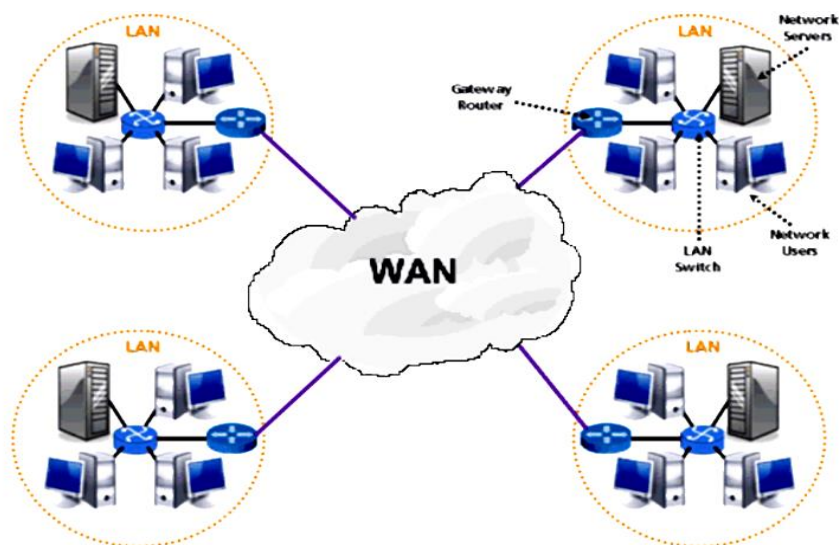


Figure 1.4. Wide Area Network (WAN) [3].

1.5. Network Architecture (Model)

Network architecture is a hierarchical set of layers and protocols for communication allowing various computer equipment to exchange information through a network. There are several network architectures like the OSI (Open Systems Interconnection) model which is the international reference architecture and the TCP/IP architecture (Transfer Control Protocol / Internet Protocol) which is the standard model caused by the evolution of the Internet it uses [5].

1.5.1. OSI Network Model

The Open Systems Interconnection model (OSI) is a conceptual model that characterizes and standardizes the internal functions of a communication system by partitioning it into abstraction layers. The model is a product of the Open Systems Interconnection project at the International

Organization for Standardization (ISO), maintained by the identification ISO/IEC 7498-1. The model group communication functions into seven logical layers. A layer serves the layer above it and is served by the layer below it. For example, a layer that provides error-free communications across a network provides the path needed by applications above it, while it calls the next lower layer to send and receive packets that make up the contents of that path [4].

a) Description of the OSI model layers

The OSI model has seven layers shown in figure 1.5 and described in the following subsections:

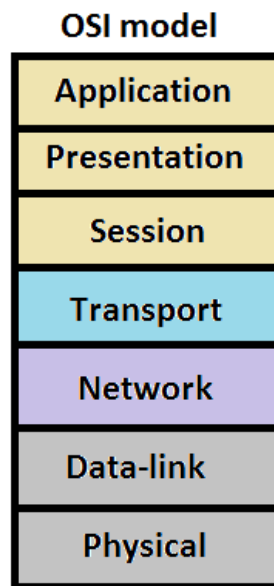


Figure 1.5. OSI model layers

- **Physical layer:** it is concerned with the transmission and reception of the unstructured raw bit stream over a physical medium. It describes the electrical/optical, mechanical, and functional interfaces to the physical medium, and carries the signals for all of the higher layers.
- **Data link layer:** it provides error-free transfer of data frames from one node to another over the physical layer, allowing layers above it to assume virtually error-free transmission over the link.
- **Network layer:** it controls the operation of the subnet, deciding which physical path the data should take based on network conditions, priority of service and other factors.
- **Transport layer:** ensures that messages are delivered error-free, in sequence, and with no losses or duplications. It relieves the higher layer protocols from any concern with the transfer of data between them and their peers. The size and complexity of a transport protocol depends on the type of service it can get from the network layer. If the network layer is unreliable

and/or only supports datagrams, the transport protocol should include extensive error detection and recovery.

- **Session layer:** allows session establishment between processes running on different stations.
- **Presentation layer:** formats the data to be presented to the application layer. It can be viewed as the translator for the network. This layer may translate data from a format used by the application layer into a common format at the sending station; then, translate the common format to a format known to the application layer at the receiving station.
- **Application layer:** The application layer serves as the window for users and application processes to access network services.

1.5.2. TCP/IP Network Model

The TCP/IP model was born out of an experimental project by the American agency DARPA (Defense Advanced Research Projects Agency) in 1969. From 1983, the TCP / IP model gradually established itself as a reference model.

a) Description of the TCP/IP model layers

The TCP/IP model has uniquely four layers summarized in figure 1.6 and described in the coming subsections [5].

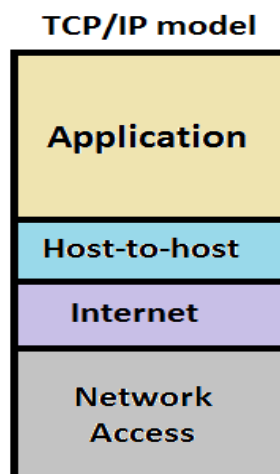


Figure 1.6. TCP/IP Model layers

- **Network Access layer:** It is the first layer of the TCP / IP protocol, providing the ability to access a physical network. It serves as an interface with the transmission medium and determines how the data should be routed.
- **Internet layer:** it is used to interconnect distant heterogeneous networks in an unconnected mode. Its role is to ensure the addressing and routing of packets in the network.
- **Transport layer:** it ensures the transmission of data and the correction of errors during the routing of data in the communication medium.
- **Application layer:** provides the functionality and means necessary to access the other layers and to facilitate the use and management of the TCP/IP network.

1.6. Machine identification

Each machine is identified by an address. Each identifier must be unique across the network. Each machine has only one address per network on which it is connected. Machines (router, gateways) which are multi-domiciled, that is to say which have several IP addresses are special cases [6].

1.6.1. MAC address

MAC (media access control) address is a device's "physical" address. It's hard-coded to the network card from the manufacturer. Represented by a 12-digit, hexadecimal number; every device in the world has a unique MAC address. MAC addresses are typically used only to direct packets from one device to the next as data travels on a network. Data link layer is the responsible layer for MAC addressing and to identify the physical address of the sender and the next device that will receive the data.

1.6.2. IP Addressing

The IP address is a unique address assigned to each computer on the Internet in a local network (i.e., there is no computer that can be found on the Internet with the same IP address). The IP address is usually displayed in 4 numbers (between 0 and 255) separated by dots [7]. For example: 204.35.129.3.

1.6.3. The different Classes of IP addresses

IP addresses are divided into classes; the most common being classes A, B, C as shown in table 1.1. There are classes D and E, but in general they are not used by end users. Each address class has a different subnet mask. To find out the class of an IP address, just look at the first byte [7].

Table 1.1. The IPv4 Classes.

Classes	Address ranges
A	1.0.0.1 to 126.255.255.254
B	128.1.0.1 to 191.255.255.254
C	192.0.0.1 to 223.255.255.254

1.6.4. Subnet mask

It allows us to define several IP addresses from an address by saying the host identifier field in two parts: subnet, host. The notion of the subnet mask is an important notion insofar as it is the mask that determines the number of machines that there may be on the network [7]. Table 1.2 gives the subnet masks of classes A, B and C.

Table 1.2. The default subnet masks for each class.

Classes	Subnet mask
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

1.6.5. DHCP protocol

Traditionally, a user could either configure the address of their device manually or had the host acquire its address dynamically through dynamic methods such as Dynamic Host Configuration Protocol (DHCP). DHCP server is the most important network infrastructures. It is considered as an essential component used to provide IP address and host configuration parameters dynamically to user devices [8].

1.6.6. DNS protocol

In the domain name server, we use a serial of simple and meaningful words that are separated by dots to replace the numerical IP address. For example, a domain name "www.ndhu.edu.tw" is the access point to the WWW server in the National Dong Hwa University, educational institution, Taiwan. Whenever you want to access the resource on this server, you only need to type "www.ndhu.edu.tw" and the domain name server will help you to resolve it into the IP address "203.64.88.52". You need not to memorize those complex numerical IP addresses [9].

1.7. Internet of things (IoT)

Internet of Things (IoT) is a system of interconnected computing devices, mechanical and digital machines, usually called smart devices, through the Internet. These devices have been assigned an IP address and have the capability to collect and transfer data over a network. The objects interact with the external environment with the help of embedded technology, which helps them in taking decisions without requiring human-to-human or human-to-computer interaction [10].

1.8. Characteristics of IoT

- **Interconnectivity:** With regard to the IoT, anything can be interconnected with the global information and communication infrastructure [11].
- **Heterogeneity:** The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks [11].
- **Dynamic changes:** The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically [11].
- **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. The ratio of communication triggered by devices as compared to communication triggered by humans will noticeably shift towards device triggered communication [11].
- **Safety:** both the creators and recipients of the IoT must design for safety. This includes the safety of personal data and the safety of physical well-being. Securing the endpoints, the networks, and the data moving across all of it means creating a security paradigm that will scale [11].

1.9. IoT Devices

An IoT device is a piece of hardware with a sensor that transmits data from one place to another over the Internet. Types of IoT devices include wireless sensors, software, actuators, and computer devices. They can be imbedded into mobile devices, industrial equipment, environmental sensors, medical devices, and more. Connected IoT devices, which convey usage and other data, can potentially provide insights that lead companies to cost reductions, efficiency gains, and new business opportunities.

- **Sensors:** Devices which converts physical parameters like temperature, motion etc... into the electrical signals. Smart sensors are the indispensable enablers of IoT.
- **Actuators:** Devices which are contrast to sensors. They transform electrical signals into physical movements. Both sensors and actuators are transducers which converts one form of energy to other. Exchange of data is the most important key factor in IoT. Hence sensors and actuators play a vital role here.
- **RFID Tags:** (Radio-frequency identification). Wireless microchips used for automatic unique identification of anything by tagging it over them. We have been seen it in credit cards, automobile ignition keys and so on. Since interconnection of things is the main goal of IoT, the RFID tags get hand shaken with IoT technology and is used to provide the unique ID for the connected “things” in IoT [13].

1.10. Applications of IoT

1.10.1. *The Smart Home*

Future smart homes will be conscious about what happens inside a building, mainly impacting three aspects: resource usage (water conservation and energy consumption), security and comfort [12].

1.10.2. *Smart Cities*

Smart city is an urban area which creates sustainable development and high quality of life. The characteristics of smart city are: encompassing economy, people, governance, mobility, environment and living. Outperforming in these key areas can be achieved through strong human or social capital and/or ICT infrastructure [12].

1.10.3. *Transport/Logistics*

In transport logistics, IoT improves not only material flow systems but also the global positioning and automatic identification of freight. It also increases energy efficiency and thus decreases energy consumption [12].

1.10.4. *E-Health*

Control and prevention are two of the main goals of future health care. Already today, people have the option of being tracked and monitored by specialists even if the patient and specialist are not in the same place. In this domain, IoT makes human interaction much more efficient because it permits not only localization, but also tracking and monitoring of patients. Providing information about the state of a patient makes the whole process more efficient, and also makes people much more satisfied [12].

1.10.5. *Smart retail*

Retail IoT realizes both customer needs and business needs: price comparison of a product, looking for other products of the same quality at lower prices; with shop promotions, giving information not only to customers but also to shops and businesses. Having this information in real time helps enterprises to improve their business and to satisfy customer needs [12].

1.10.6. *Smart farming*

The Internet of Things could revolutionize the way farmers work. Smart farming will become the important application field in the predominantly agricultural-product exporting countries.

1.11. Conclusion

In conclusion, a network is two or more computers connected together using a telecommunication system for the purpose of communicating and sharing resources.

Internet of Things is a new revolution of the Internet and it is a key research topic for researcher in embedded, computer science and information technology area due to its very diverse area of application and heterogeneous mixture of various communications and embedded technology in its architecture.

2

SMART HOME EQUIPMENT AND INTERCONNECTION

A network includes several IT equipment located remotely one by one compared to others. The first task to be implemented to constitute the network is the transmission of information from one equipment to another. The transmission media have the role of conveying information between any two distant elements. In this chapter, we will see the different types of equipment and supports transmission used.

2.1. Network Equipment

The interconnection of networks can be local: the networks are on the same geographic site. In this case, standard equipment (hubs, switch, router, etc....) is sufficient to carry out physical bonding. Interconnection can also concern remote networks. It is then necessary to connect these networks by a telephone link (modems).

2.1.1. Network hub

A **network hub** serves as a connection point for all devices in a LAN (see figure 2.1). Hubs are OSI Layer 1 devices and have no concept of Ethernet frames or addressing. They have no way of distinguishing which port a signal should be sent to instead; an electrical signal is broadcast to every port. All nodes on the network will receive data, and the data will eventually reach the correct destination, but with a lot of unnecessary network traffic [14].

Actually, for our project we didn't use hubs, we used switch instead to avoid collision and any unnecessary traffic that can cause delays.



Figure 2.1. Network Hub [14]

2.1.2. Network switch

It is a mean that connects devices on a LAN. An Ethernet switch usually works at the Data link layer of the OSI model (Layer 2). It manages the flow of data across a network by inspecting the incoming frame's destination MAC address and forwarding the frame only to the host for which the message was intended. Each switch has a dynamic table (called the **MAC address table**) that maps MAC addresses to ports. With this information, a switch can identify which system is sitting on which port and where to send the received frame [14].

There is another kind of switches that works at the network layer of the OSI model (Layer 3). The switch is used to allow the two servers that are needed, to connect to the same network. A switch is shown in figure 2.2.



Figure 2.2. Network Switch [14].

2.1.3. Network Router

Router (shown in figure 2.3) is a network device that connects different computer networks by routing packets from one network to another. This device is usually connected to two or more different networks. When a data packet comes to a router port, the router reads the address information in the packet to determine out which port the packet will be sent. For example, a router provides us with internet access by connecting our LAN with the Internet.



Figure 2.3. Network Router [14]

A router is considered a Layer 3 device of the OSI model because its primary forwarding decision is based on the information of the OSI Layer 3 (the destination IP address). If two hosts from different networks want to communicate with each other, they will need a router between them [14].

For this project, a router is used to create the network and connect several LANs together; therefore, connect our LANs to the Internet.

2.1.4. Network modem

The modem is defined as an abbreviation for modulator-demodulator, a modem is a hardware device that allows a computer to send and receive information over telephone lines or coaxial cables. When sending a signal, the device converts ("modulates") digital data to an analog signal, and transmits it over a telephone line. Similarly, when an analog signal is received, the modem converts it back ("demodulates" it) to a digital signal [15].

Both DSL (Digital Subscriber Line) and cable modems are common home networking broadband connection technologies. Figure 2.4 is an example of cable and DSL modems. DSL uses a sophisticated modulation scheme to pack data onto copper wires. DSL is sometimes referred to as a last-mile technology because it is used only for connections from a telephone switching station to a home or office, not used between switching stations. Through the use of a cable modem, we can have a broadband Internet connection that is designed to operate over cable TV lines.



Figure 2.4. Cable and DSL modems

2.1.5. Server

A server is a software or hardware device that accepts and responds to requests made over a network. Figure 2.5 gives an illustration of IBM network servers. The device that makes the request, and receives a response from the server, is called a *client*. On the Internet, the term "server" commonly refers to the computer system which receives a request for a web document and sends the requested information to the client. Some examples of servers are [15]:

- **Web Server:** A computer or collection of computers used to deliver web pages and other content to multiple users.
- **File Server:** A computer on a network that is used to provide users on a network with access to files.
- **DNS Server:** Short for Domain Name System, a DNS is a service that receives a request containing a domain name hostname and responds with the corresponding IP address.
- **Database Server:** A database server is a computer system that provides other computers with services related to accessing and retrieving data from a database.



Figure 2.5. IBM Network Servers.

In our project, three servers are needed: the first one for DNS, the second one for the IoT service, and the last one for the 3G/4G provider.

2.1.6. Cloud

Cloud networking, or cloud-based networking, is when some or all of an organization's networking resources are hosted in the cloud. This may refer to either a public or private cloud. It is based on Cloud computing, which is a term used to describe services provided over a network by a collection of remote servers as shown in figure 2.6. This abstract "cloud" of computers provides massive, distributed storage and processing power that can be accessed by any Internet-connected device running a web browser [15].



Figure 2.6. Cloud illustration [15]

2.1.7. Cell Tower

Alternatively referred to as cell site, a cell tower is a tall tower equipped with electronics along with an antenna that transmits data to and cellular phones as the one shown in figure 2.7. These towers are often located throughout a city in high areas and at the top of buildings [15].

Cell towers are grouped in geographical locations where the population density is high and there are likely to be large numbers of cell phone users. This helps in avoiding saturation of the available capacity, which could result in busy signals and unhappy consumers. Cell phones are designed to be aware of the nearest tower. This is shown to the user in the form of signal strength, which represents the connectivity strength between the user's location and the nearest tower providing the service.



Figure 2.7. Cell Tower

For our project, the 3G/4G client will be connected to this Cell tower; the tower itself will be connected to the internet through a server.

2.2. Network Media

Media is the actual physical environment through which data travels as it moves from one component to another, and it connects network devices. The most common types of network media are twisted-pair cable, coaxial cable, fiber-optic cable, and wireless. Each media type has specific capabilities and serves specific purposes.

Understanding the types of connections that can be used within a network provides a better understanding of how networks function in transmitting data from one point to another [16].

2.2.1. Twisted-Pair Cable

Twisted-pair is a copper wire-based cable that can be either shielded or unshielded. Twisted-pair is the most common media for network connectivity [16].

- **Unshielded twisted-pair (UTP)** cable is a four-pair wire. Each of the eight individual copper wires in UTP cable is covered by an insulating material. Also, the wires in each pair are twisted around each other. The advantage of UTP cable is its ability to cancel interference because the twisted-wire pairs limit signal degradation from electromagnetic interference (EMI) and radio frequency interference (RFI). To further reduce crosstalk between the pairs in UTP cable, the number of twists in the wire pairs varies. Several categories of UTP cable exist up to six categories [16].
 - **Shielded twisted-pair (STP)** cable combines the techniques of shielding and the twisting of wires to further protect against signal degradation. Each pair of wires is wrapped in a metallic foil. The four pairs of wires are then wrapped in an overall metallic braid or foil. Specified for use in Ethernet network installations, STP reduces electrical noise both within the cable (pair-to-pair coupling, or crosstalk) and from outside of the cable (EMI and RFI) [14].
- UTP, as well as shielded twisted-pair (STP) cables (shown in figure 2.8), must follow precise specifications as to how many twists or braids are permitted per meter [16].

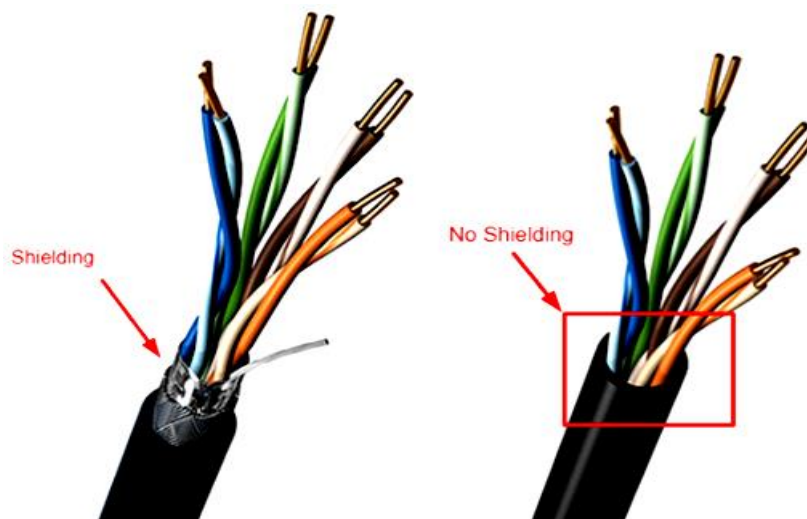


Figure 2.8. STP and UTP Cables [16]

2.2.2. Coaxial

cable consists of a hollow outer cylindrical conductor that surrounds a single inner wire conducting element. The single inner wire located in the center of a coaxial cable is a copper conductor, surrounded by a layer of flexible insulation. Over this insulating material is a woven

copper braid or metallic foil that acts both as the second wire in the circuit and as a shield for the inner conductor. This second layer, or shield, can help reduce the amount of outside interference. An outer jacket covers this shield as shown in figure 2.9.



Figure 2.9. Coaxial Cable

Coaxial cable can be laid over longer distances than twisted-pair cable. For example, Ethernet can run approximately 100 meters using twisted-pair cable, but 500 meters using coaxial cable.

Coaxial cable offers several advantages for use in LANs. It can be run with fewer boosts from repeaters, which regenerate the signals in a network so that they can cover greater distances between network nodes than either STP or UTP cable. Coaxial cable is less expensive than fiber-optic cable, and the technology is well known. It has been used for many years for all types of data communication [16].

2.2.3. Fiber-optic

cable is a networking medium capable of conducting modulated light transmission. The Fiber-optic cable used for networking consists of two fibers encased in separate sheaths. Each optical fiber is surrounded by layers of protective buffer material: usually a plastic shield, then a plastic such as Kevlar, and finally, an outer jacket that protects the entire cable (see figure 2.10). Buried fiber-optic cables are required by codes, a stainless-steel wire is sometimes included for added strength.

The light-guiding parts of an optical fiber are called the *core* and the *cladding*. The core is usually very pure glass with a high index of refraction. When a cladding layer of glass or plastic with a low index of refraction surrounds the core glass, light can be trapped in the fiber core. This process is called *total internal reflection*, and it allows the optical fiber to act as a light pipe, guiding light for long distances, even around bends. Fiber-optic cable is the most expensive of the three types discussed, but it supports higher rate line speeds. Two types of fiber-optic cable exist:

- **Single-mode**—fiber-optic cable allows only one mode (or wavelength) of light to propagate through the fiber. This type of cable is capable of higher bandwidth and greater distances than multimode and is often used for campus backbones. Single-mode cable uses lasers as the light-generating method and is more expensive than multimode cable. The maximum cable length of single-mode cable is +60 km.
- **Multimode**—fiber-optic cable allows multiple modes of light to propagate through the fiber. Multimode cable is often used for workgroup applications, using light-emitting diodes (LEDs) as light-generating devices. The maximum length of multimode cable is 2 km.

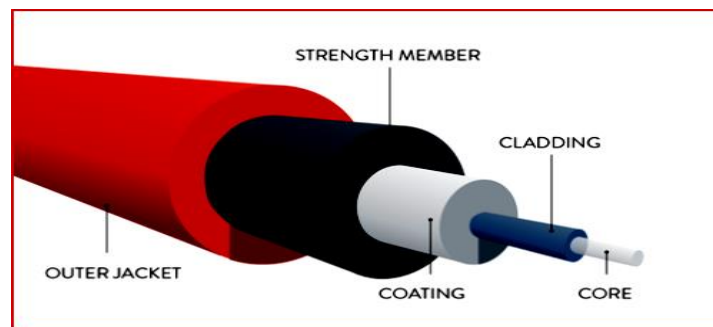


Figure 2.10 Fiber Optic Cable

Although fiber-optic cable is more expensive, it is not susceptible to EMI and is capable of higher data rates than any of the other types of networking media discussed here. Fiber-optic cable is also more secure because it does not emit electrical signals that could be received by external devices [16].

2.2.4. Wireless communication

networks are becoming increasingly popular, and they utilize a different type of technology. Wireless communication uses radio frequencies (RFs) or infrared waves to transmit data between devices on a LAN. For wireless LANs, a key component is a wireless hub, or access point, used for signal distribution. To receive the signals from the access point, a PC or laptop needs to install a wireless adapter card, or wireless network interface card (NIC) [16].



Figure 2.11. Cisco Wireless Access Point

2.3. Network Connectors

2.3.1. Connector definition

A connector is a device that terminates a segment of cabling or provides a point of entry for networking devices such as computers, hubs, and routers. Several types of connectors can be used in LANs, depending on the requirements for the network and the type of Ethernet to be implemented. These connectors also vary depending on the type of media that is installed [17].

2.3.2. RJ connectors

RJ (Registered Jack) Connectors are a family of push-and-click connectors for twisted-pair wiring in telephone and network wiring. RJ stands for Registered Jack. RJ types define both a jack and receptacle (female) and a plug (male) type of connector (see figure 2.12 (a)), the most used type is RJ-45 and RJ-11 [17].

2.3.3. BNC (Bayonet Neill–Concelman) connector

It is used for connecting thinnet coaxial cabling to various networking components (see figure 2.12(b)) [17].

2.3.4. SC and ST connectors

SC/ST connectors, shown in figure 2.12 (c), are used for connecting fiber-optic cabling to networking devices SC stands for Subscriber Connector and ST stands for Straight Tip [17].



Figure 2.12. Network connectors

2.4. Smart Home Required Equipment

The home considered in our study consists of 5 parts: Garage, Door, Room, Garden and Kitchen. Each part will be discussed separately in the following subsections.

2.4.1. Garage

The garage door is equipped with an **RFID Reader** (see figure 2.13(c)) which allows checking the cards authority of the person willing to access into the garage.

Moreover, the garage entry is outfitted with a **motion sensor** (shown in figure 2.13 (a)) that detects motion and a **photo-sensor** (shown in figure 2.13(b)) that detects light within the room where the sensor is placed and a **Smart LED** to light-on the home when necessary. These pieces of equipment are connected to a **micro-controller unit** (see figure 2.13(d)) that turns the LED ON/OFF and opens the garage door depending on the given data from the sensors and the RFID Reader.



(a) Motion Sensor (b) Photo Sensor (c) RFID Reader + RFID Card (d) Micro-controller unit

Figure 2.13. Garage required pieces of equipment

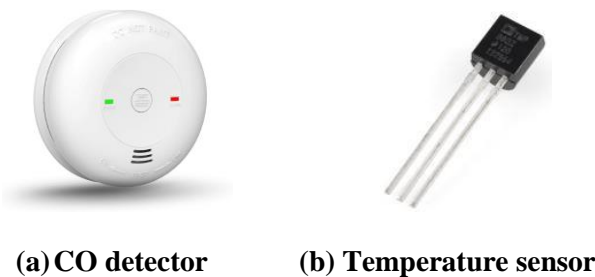
2.4.2. Door

This part has the same components as that are used in the garage. Another RFID Reader with other cards, a motion sensor, a photosensor and another micro-controller.

2.4.3. The Room

The Room is divided into two sections: Carbon monoxide monitoring and Auto temperature monitoring as explained below.

- **Carbon monoxide monitoring:** this section is composed of a **carbon monoxide detector**, illustrated in figure 2.14(a), that detects the level of carbon monoxide and accordingly the window and fan will be ON/OFF.



(a) CO detector (b) Temperature sensor

Figure 2.14. Room required equipments

- **Auto temperature monitoring:** this section is composed of an Air Conditioner (AC), a furnace, and a **temperature sensor** (shown in figure 2.14(b)) which is a sensor that outputs temperature in Celsius. These devices are controlled by a Micro-Controller Unit (MCU) to make decisions about the status of the AC and the furnace.

2.4.4. The Garden

The garden is divided into three parts: water monitoring, fire monitoring, and CO level monitoring.

- **Water monitoring:** this section is composed of **lawn sprinklers**, shown in figure 2.15(a), which sprinkles water and a water level sensor which measures the amount of water.
- **Fire monitoring:** this section is composed of a fire detector that detects IR in the range of fire and a **Fire Sprinkler**, shown in figure 2.15(b) that puts out the fire.



(a) Lawn sprinkler



(b) Fire sprinkler

Figure 2.15. Garden required pieces of equipment

2.4.5. Kitchen

The kitchen design is also divided into two sections: Auto fire monitoring and Auto smoke monitoring described in the coming subsections.

- **Auto fire monitoring:** this section is almost the same as the one in the garden; also, a **siren** is needed to make an alert in case there is a fire.
- **Auto smoke monitoring:** this section is composed of a **smoke sensor**, shown in figure 2.16, which detects the environment variable smoke level. It is attached to the Micro-Controller Unit that is used to control the Kitchen window (open/close), based on the smoke level.



Figure 2.16. Smoke sensor

2.5. Conclusion

IoT is a network in which all physical objects are connected to the internet through network devices or routers and exchange data. Nowadays, almost all the devices are still not smart, or they cannot be connected to the internet, even if some companies are working on that for enough time; in our country, we still not able to implement this kind of project.

3

THE CONFIGURATION OF LANs, WAN, AND THE IOT DEVICES

The purpose of this chapter is to describe the methods and the steps on how this work is conducted. Moreover, the project process and methodologies and the practical steps are explained.

Due to the extra complexity in having real hardware such as microcontrollers, sensors, and actuators, it was decided to utilize an IoT simulator. A choice was made to use the Cisco Packet Tracer simulation tool.

3.1. Cisco Packet Tracer Overview

Cisco Packet Tracer is a powerful network simulation program that allows students to experiment with network behavior and ask “what if” questions. As an integral part of the Networking Academy comprehensive learning experience, Packet Tracer provides simulation, visualization, authoring, assessment, and collaboration capabilities to facilitate the teaching and learning of complex technology concepts [18].

The Cisco Packet Tracer is a simulation program that provides a network lab environment for users to perform Cisco operations or applications without requiring the use of any physical machine or vehicle. The tool offers an extensive set of hardware and cabling that allows students to set up from a basic to very complex network. It also educates how to troubleshoot network-related problems, as the tool also includes realistic features for debugging [19].

From version 7.0 Cisco also introduced IoT functionalities in the tool, allowing students to practice by setting up IoT devices and IoT automation. Also, a possibility for a lower level IoT simulation using a single-board computer (SBC) and sensors were offered in the same release.

This project work was only focusing on delivering IoT simulations utilizing Cisco Packet Tracer. It was not in the scope of this project to evaluate or compare other IoT simulator available. This project work is written with the latest release available version which is 7.3.0.

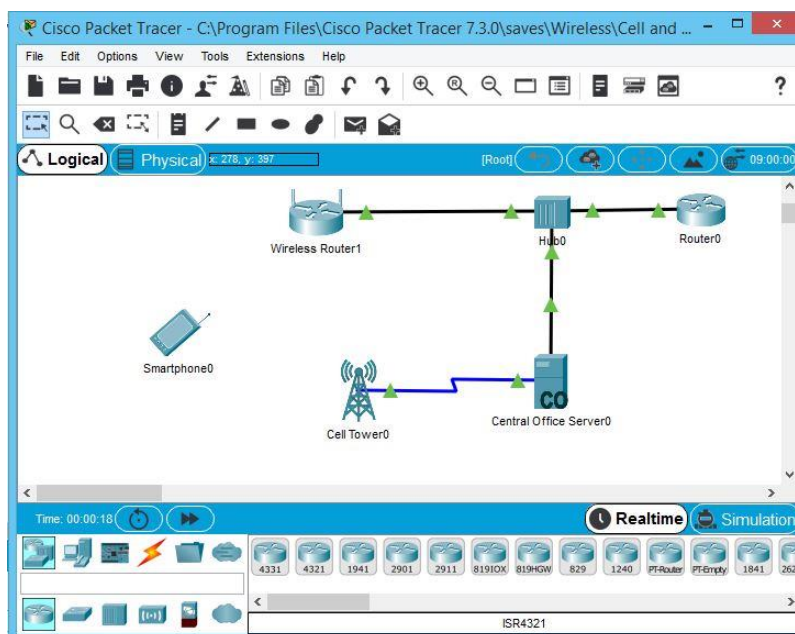


Figure 3.1. Cisco Packet Tracer interface.

3.2. Network Layout

To implement a smart home that can be automatically and remotely monitored from a 3G/4G client, the network was logically separated in four areas: home network, Internet provider Cloud, Service provider (SP) network, and 3G/4G Provider network.

Figure 3.2 illustrates the project network topology implemented in the IoT simulator chosen (Cisco Packet Tracer).

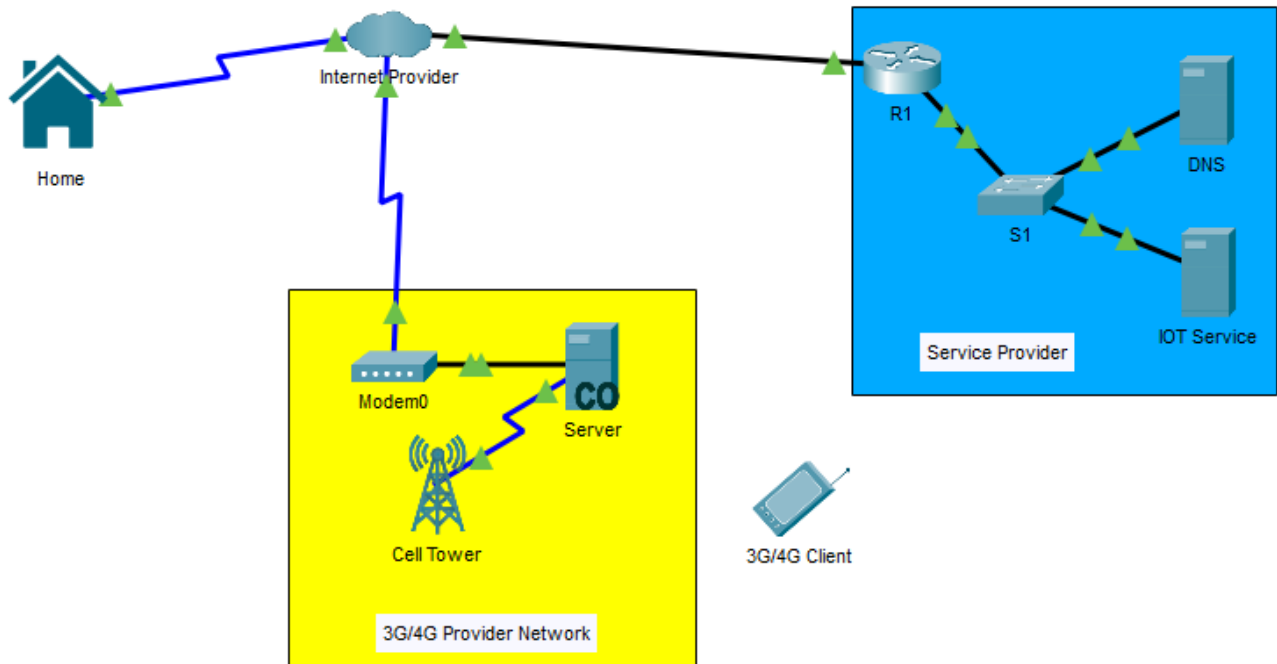


Figure 3.2. Our smart home network topology.

3.3. IPv4 Addresses of the system

Table 3.1 below shows all devices' IP addresses that will get after configuration.

Table 3.1. IPv4 Addresses of all devices.

Section	Device	Interface	IPv4 Address
Service Provider	Router R1	FastEthernet0/0	192.168.1.1
		FastEthernet0/1	10.0.0.1
	DNS server	FastEthernet0	192.168.1.2
	IoT sever	FastEthernet0	192.168.1.3
3G/4G Provider	Central Office Server	Backbone	Dynamic address from 10.0.0.0/8 except (10.0.0.1)
		Cell Tower	172.168.1.1
	3G/4G clients	3G/4G Cell	Dynamic address from 172.168.1.0/24 except (172.168.1.1)
Smart Home	Home Gateway	Internet	Dynamic address from 10.0.0.0/8 except (10.0.0.1)
		LAN	192.168.3.1
	Laptop	Wireless0	192.168.3.2
	IoT Devices	Wireless0	Dynamic address from 192.168.3.0/24 except (192.168.3.1 and 192.168.3.2)

3.4. Internet configuration

We need to set up all configurations for all devices, to achieve the appropriate communication between the home devices and servers. And allow the 3G/4G client to access the home.

3.4.1. Internet provider

To provide internet to all sections, a cloud that acts as a cable forwarder is needed from port Coax8 to port FastEthernet9 and from port Coax7 to port FastEthernet9 as shown in figure 3.3(a). Whereas, figure 3.3(b) shows the configuration TAB of the CLOUD.

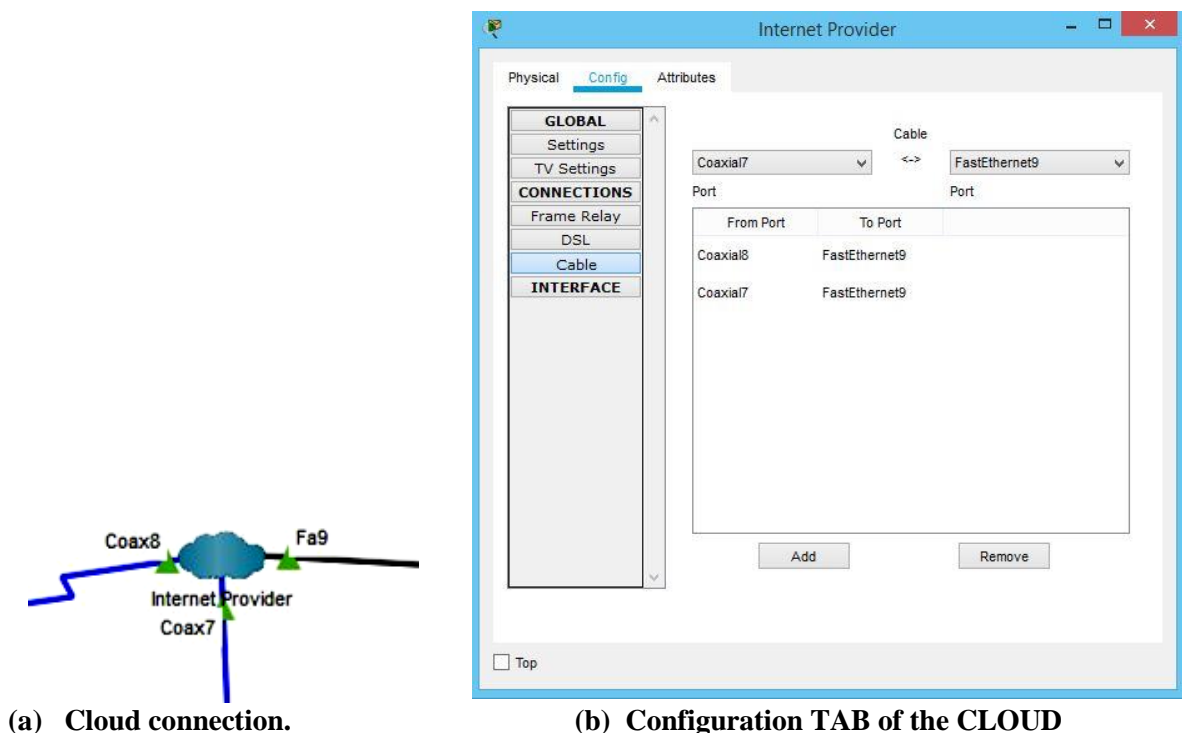


Figure 3.3. Internet provider setting.

In the coming sections, an interface will be connected to the internet provider, the WAN IP address to be 10.0.0.0/8 is chosen, and it can support more than 16 million sections. Addresses are provided by a DHCP server which is the router R1 in the Service provider section.

3.4.2. Service provider

3.4.2.1. Router (R1)

This router connects the service provider LAN that is connected to its FastEthernet0/0 (Fa0/0) with the other LANs, through a WAN connected to its FastEthernet0/1 (Fa0/1) interface. The following commands are written to the CLI (command-line interface) TAB of R1:

- **For FastEthernet0/0:**

```
R1(config)#interface fa0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#no shutdown
```

- **For FastEthernet0/1:**

```
R1(config)#interface fa0/1
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
```

This interface address has to be set as a default router for the DHCP server to provide addresses from 10.0.0.0/8 network with a DNS server 192.168.1.2 as follow:

```
R1(config)#ip dhcp excluded-address 10.0.0.1
R1(config)#ip dhcp pool Dhiya
R1(dhcp-config)#network 10.0.0.0 255.0.0.0
R1(dhcp-config)#default-router 10.0.0.1
R1(dhcp-config)#dns-server 192.168.1.2
```

Finally, each device connected to this network and enables the DHCP protocol, must get an address from the pool “Dhiya” (10.0.0.2 to 10.255.255.254).

3.4.2.2. Switch (S1)

The switch is self-configured; it is used just to connect the Fa0/0 interface from R1 to multiple devices.

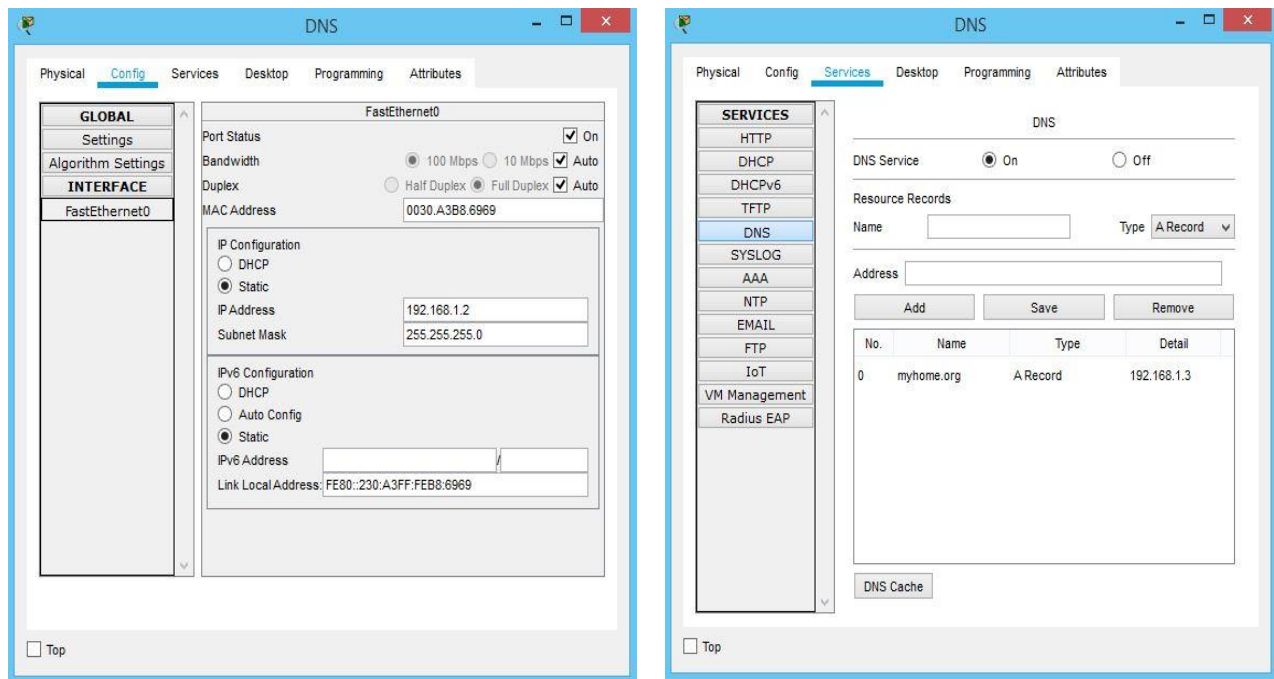
3.4.2.3. DNS

- **Network Configuration**

First, we configure a static IPv4 address for the server. IPv6- will not be used. The server is connected to the network “192.168.1.0” with the subnet mask 255.255.255.0, through the FastEthernet0 interface. The default gateway address is “192.168.1.1”. So, “192.168.1.2” is set as an address and the port status is checked to enable the interface as illustrated in figure 3.4(a). We go to settings in the same TAB and add the DNS and gateway IP addresses.

- **Domain Name Translation**

To translate the IoT server IP address “192.168.1.3” to the domain “myhome.org”, we go to Services TAB of the server, then DNS and we set DNS service to ON. Then we add the domain name and the IP address as shown in the figure 3.4(b).



(a) DNS Network configuration.

(b) Domain Name Translation

Figure 3.4. DNS configuration

3.4.2.4. IoT service

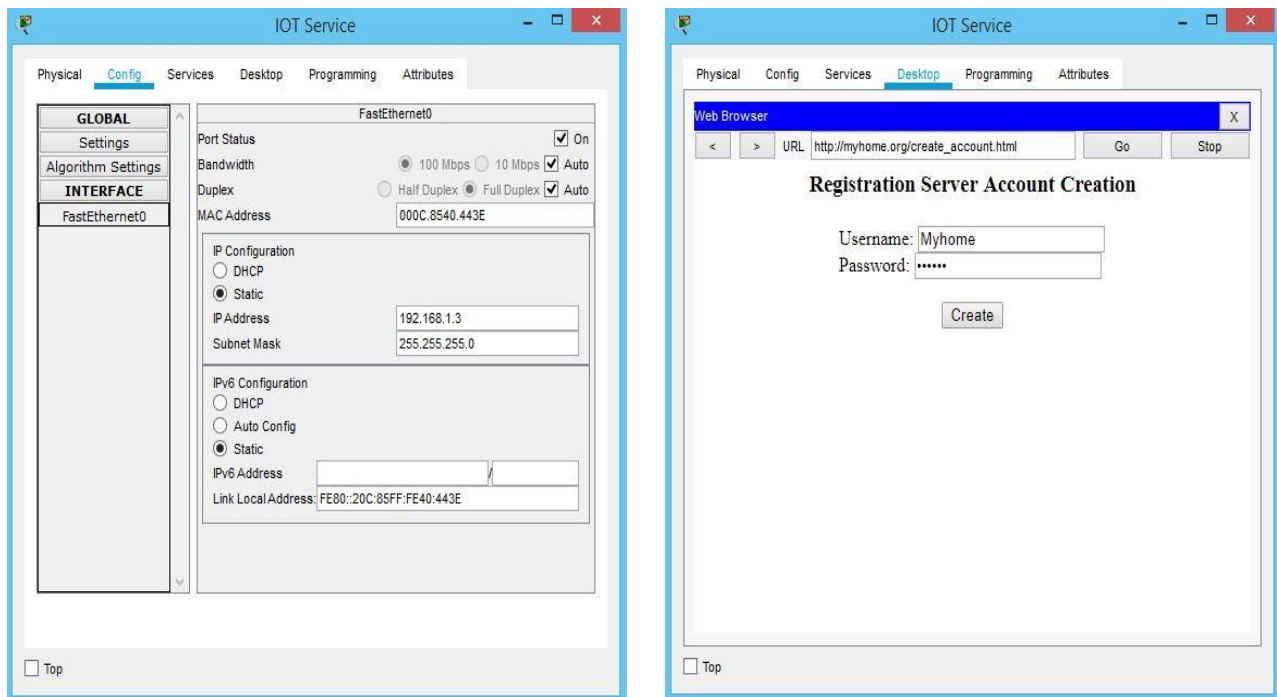
- **Network configuration**

For interface FastEthernet0, it is also connected to the network “192.168.1.0” with the subnet mask 255.255.255.0. The default gateway address is “192.168.1.1”. We give it “192.168.1.3” as an address and check the port status to enable the interface as illustrated in figure 3.5(a). We go to settings in the same TAB and add the DNS and gateway IP addresses.

- **Setting up the IoT service**

To make this server work as an IoT server, the IoT service has to be enabled from services TAB. Then, an account must be created in the server, by starting a web browser from the desktop of the server itself and access the IoT server address “192.168.1.3” or the domain name “myhome.org”. Since we do not already have an account, we need to click “sign up now”.

For us, the account username is “Myhome” as shown in figure 3.5(b).



(a) IoT service network configuration.

(b) Setting up the IoT service

Figure 3.5. IoT service configuration

- **The way to access our IoT account**

Finally, Since the IoT server is connected to the Internet. We can access any account created in the server following this procedure:

- 1- Start a web browser from any device that is connected to the internet.
- 2- Type the IP address of the server “192.168.1.3” or the domain name “myhome.org”.
- 3- Use the correct username and password.
- 4- Then click “Sign In”.

3.4.3. 3G/4G provider network

3.4.3.1. Cell Tower

We can only change the provider name so that any 3G/4G client can connect with the provider by typing the appropriate name. The name chosen was INELEC.

3.4.3.2. Central Office (CO) Server

It has two interfaces: the backbone to connect the provider to the WAN, and a coaxial interface to connect the server with the Cell Tower.

- **Backbone interface:** this interface is connected to the internet provider. So, it will get a dynamic address once we enable the DHCP protocol as shown in figure 3.6.

The screenshot shows a window titled "Backbone Settings". Under "IP Configuration", the "DHCP" radio button is selected. Below this, there are four input fields: "IP Address" with the value "10.0.0.4", "Subnet Mask" with "255.0.0.0", "Default Gateway" with "10.0.0.1", and "DNS Server" with "192.168.1.2".

Figure 3.6. Backbone interface DHCP enabling.

- **Cell Tower interface:** the CO server works also as a DHCP server for all devices that are connected to the Cell Tower. The configuration on figure 3.7 allows the CO server to give addresses from 172.16.1.2 to 172.16.1.254.

The screenshot shows a window titled "Tower Interface". Under "IP Configuration", there are two input fields: "IP Address" with the value "172.16.1.1" and "Subnet Mask" with "255.255.255.0".

Figure 3.7. Cell tower interface configuration.

3.4.3.3. Modem

Once the Ethernet cable and the coax cable are connected, the modem starts working, without need to any extra configuration.

3.4.3.4. 3G/4G Client

By typing the name of the 3G/4G provider “INELEC”, the phone will be connected to the tower. Hence, the CO server will assign an address to the Client from the network 172.16.1.0/24, as illustrated in figure 3.8 below.

The screenshot shows a window titled "3G/4G Cell1". At the top, "Port Status" is set to "On" with a checked checkbox. Below that, "Provider Name" is set to "INELEC". Under "IP Configuration", there are two input fields: "IP Address" with the value "172.16.1.100" and "Subnet Mask" with "255.255.255.0". A "DHCP Refresh" button is located at the bottom right.

Figure 3.8. 3G/4G client configuration.

The default gateway will be “172.16.1.1” and the DNS server provided by the service provider is “192.168.1.2”. The settings can be verified from the Config TAB of the Client.

Following the explained steps, the 3G/4G Client will be connected to the internet and get the authority to access the IoT server. If this client has the right username and password, it will be able to access the account; also, it will be allowed to do some changes.

3.4.4. Home

As already described in section 2.4 of chapter 2, our home consists of 5 parts: Garage, Door, Room, Garden and Kitchen. Figure 3.9 shows the topology of our home design created using packet tracer software, and the way these devices are connected will be explained in the coming subsections.



Figure 3.9. Home topology.

3.4.4.1. Home Gateway:

The home gateway has three interfaces: Internet, LAN and Wireless interface. Each one is described below.

- **Internet interface:** it is the interface that connects the home to the internet through the Modem. Since this interface is connected to the Internet provider. Once we choose DHCP from the internet settings TAB, the interface will get a dynamic IP address from the pool “Dhiya”, a default gateway is “10.0.0.1” and the DNS server is “192.168.1.2” as shown in figure 3.10.

Internet Settings	
IP Configuration	
<input checked="" type="radio"/> DHCP	
<input type="radio"/> Static	
IP Address	10.0.0.3
Subnet Mask	255.0.0.0
Default Gateway	10.0.0.1
DNS Server	192.168.1.2

Figure 3.10. Internet interface of the home gateway.

- **LAN interfaces:** In this project, the appliances have wireless interfaces and the network of the home is WLAN. We didn't use Ethernet interfaces.

But, from this TAB, we have to create a LAN with an IP address and a subnet mask. We chose "192.168.3.0/24" to be the WLAN. It can support 253 devices. The home gateway is by default a DHCP server.

- **Wireless interface:** from the wireless interface configuration TAB shown in figure 3.11, a personal SSID and password can be set. To make things easy in the simulation, a password wasn't set, but practically, it should be done. The Home gateway SSID is set to "HomeGateway". It will be used for all devices.

Wireless Settings	
SSID	HomeGateway
2.4 GHz Channel	6 - 2.437GHz
Coverage Range (meters)	250,00
Authentication	
<input checked="" type="radio"/> Disabled	<input type="radio"/> WEP WEP Key
<input type="radio"/> WPA-PSK	<input type="radio"/> WPA2-PSK PSK Pass Phrase
<input type="radio"/> WPA	<input type="radio"/> WPA2
RADIUS Server Settings	
IP Address	
Shared Secret	
Encryption Type	Disabled

Figure 3.11. Home gateway wireless interface configuration.

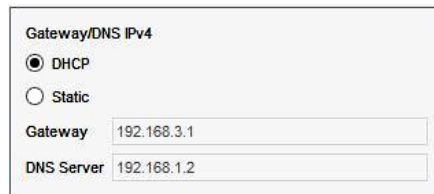
3.4.4.2. Laptop

Once we type "HomeGateway" in the SSID field on the wireless interface configuration TAB, the laptop physically connects to the home gateway. If we choose DHCP, the home gateway will give the laptop a dynamic address in the range "192.168.3.2" to "192.168.3.254" as illustrated in figure 3.12.

IP Configuration	
<input checked="" type="radio"/> DHCP	
<input type="radio"/> Static	
IP Address	192.168.3.101
Subnet Mask	255.255.255.0

Figure 3.12. Laptop wireless configuration.

Finally, the laptop will be connected to the internet. It will get the DNS server address from the home gateway and will have the authority to access the IoT server for local management as shown in figure 3.13.



Gateway/DNS IPv4

DHCP

Static

Gateway: 192.168.3.1

DNS Server: 192.168.1.2

Figure 3.13. Laptop Gateway and DNS configurations.

If the laptop doesn't have a wireless interface, open the physical TAB, turn the laptop off, remove the NIC that is placed before, add the wireless NIC "PT-LAPTOP-NM-1W" and turn on the laptop.

3.5. Configuring the IoT devices

3.5.1. Wireless interface

By default, IoT devices in Cisco Packet Tracer have Ethernet NIC, which needs a cable to connect to the home gateway, so we need to change the NIC for all devices to allow them to connect via Wireless interface.

The procedure to add a wireless NIC:

- 1- Open the advanced list of TABs of the device (see figure 3.14).
- 2- Go to "I/O Config" TAB.
- 3- Any device can support two NICs. For Network adapter, choose "PT-IOT-NM-1W".

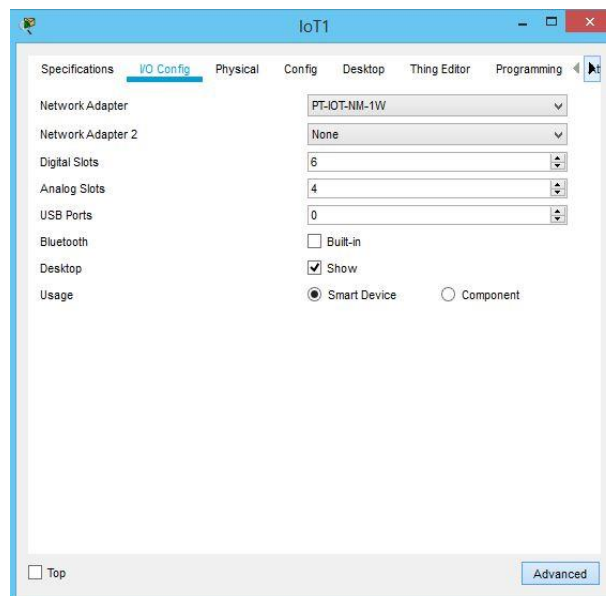


Figure 3.14. I/O Config TAB of an IoT device.

To register a device in a server, the address of the server, account username and password are needed. All devices must use the same IoT credentials, the same credentials were also used by the homeowner for passing the authentication when connecting via browser to the IoT Server.

From the config TAB of the device, in the IoT server field, select “Remote Server” (see figure 3.17); then, enter the information needed and then press “connect”.

Figure 3.17. IoT server field in the Config TAB of the device.

As said before, to monitor this home there are two ways: from the laptop inside the home for local monitoring, or from a 3G/4G client outside, or far away from home. Both methods have the same authorities and the same user interface for all the coming parts.

If the homeowner uses the laptop browser to access the IoT server, once a user is connected to “myhome.org”, it can visualize the status of the IoT devices registered in the IoT server. Figure 3.18 shows some of the appliances that will be discussed later.

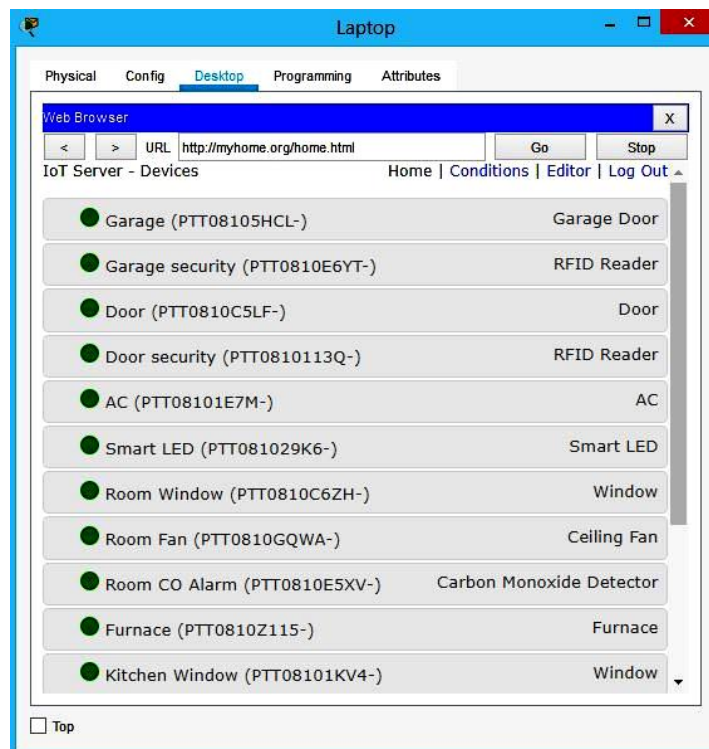


Figure 3.18. IoT device status for the devices registered in the IoT server.

This TAB (shown in figure 3.18) allows the owner to supervise some devices, whether they are working or not, and see some devices that provide useful information. From this TAB also, the owner can interact with devices directly, if the device has the property of the direct-use.

3.6. Interaction between Devices

3.6.1. Garage

For the system shown in figure 3.19, only the RFID (Garage security), Garage, and Smart LED that can be registered in the IoT server. To manage the security of the garage using the RFID we need to set some conditions in the IoT server. Hence, two cards are used: one has an authorized ID and the other has an unauthorized ID. Their IDs are numbered by 1 and 2 respectively.

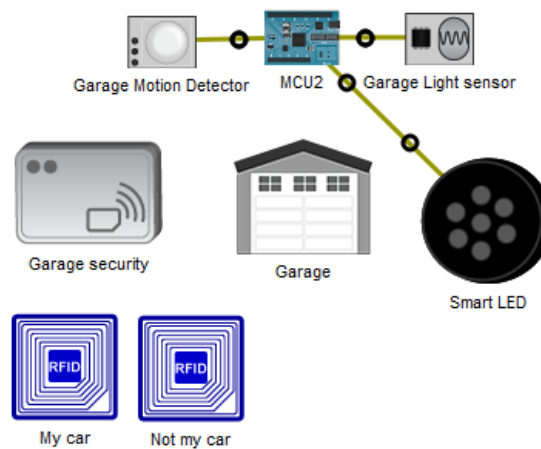


Figure 3.19. Garage security and lighting system.

The garage security and lighting algorithm can be resumed in the following four steps (this algorithm actions are also summarized in Table 3.2):

- The RFID reader works if a card is close enough to it.
- If an authorized ID card is read, the microcontroller MCU2 will send a “valid” signal to the server; else, it will send “Invalid” signal.
- If no card is close to RFID reader, MCU2 keeps sending a “waiting” signal.
- If the server receives a “valid” signal, it opens the garage for 30 seconds and closes it back. Otherwise, the garage stays closed.

For the garage lighting, the light sensor and the motion detector are connected to slot D0 and D1 of MCU2 respectively and the Smart LED is connected to slot D2 using IoT cables. A code must be typed in the MCU2 programming TAB, Python is used for MCU2 (see Appendix A).

Table 3.2. Garage security system operations.

Function Name	Condition	Action
GarageValid	Garage security status is valid	Set Garage On to true
GarageNotValidOrWaiting	Match any: -Garage security status is Invalid -Garage security status is Waiting	Set Garage On to false
GarageAuthorizedCard	Garage security Card ID=1	Set Garage security status to Valid
GarageUnauthorizedCard	Garage security Card ID=2	Set Garage security status to Invalid

The LED is controlled according to the data received from the two sensors; the following algorithm is created to manage that:

- If there is sunlight, whether there is motion or not, the LED does not light up.
- If there is no sunlight, the MCU checks the motion detector sensor values:
 - o If there is motion, it sends a “HIGH” signal to the Smart LED for 35 seconds.
 - o Else, it keeps sending a “LOW” signal for the Smart LED.
- MCU2 keeps looping to check every 3 seconds (see the complete program given in appendix A).

Alternatively, the garage can be opened manually. First, the previous conditions given in table 3.1 must be disabled; thus, the garage can be opened from the IoT server account. We can also check the Smart LED brightness as illustrated in figure 3.20.

**Figure 3.20. Check the Smart LED brightness.**

3.6.2. Door

The door's security and lighting system design is approximately the same as the garage's design. The door will not be opened and closed; it will be automatically locked and unlocked using RFID reader (named Door security in figure 3.21).

The conditions that are made in the server are the same as the conditions for the garage, but instead of opening and closing the door, the server unlocks and locks it.

We used cards different than the cards that are used for garage security, 10 is the ID of the authorized card, and 20 is for the rejected one. Table 3.3 summarizes the door security and lighting system operations.

Table 3.3. Door security system operations.

Function Name	Condition	Action
DoorValid	Door security status is valid	Set Door Lock to Unlock
DoorNotValidOrWaiting	Match any: -Door security status is Invalid -Door security status is Waiting	Set Door Lock to Lock
DoorAuthorizedCard	Door security Card ID=10	Set Garage security status to Valid
GarageUnauthorizedCard	Door security Card ID=20	Set Garage security status to Invalid

MCU4, shown in figure 3.21, is responsible of door lighting, it sends "HIGH" to the Smart LED whenever there is no sunlight and there is motion close to the door. The microcontroller program is the same as the one used for the garage lighting, the difference is only on the ON time, for the door lighting only 15 seconds are needed to turn the Smart LED back OFF.

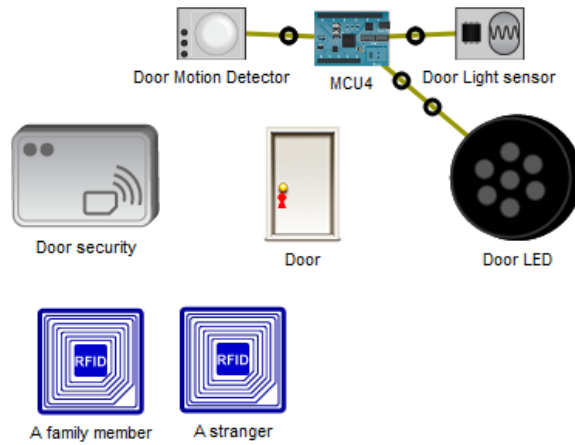


Figure 3.21. Door's security and lighting system.

Besides, we can manually monitor the door, we can lock and unlock it. Moreover, we can see if it is open or closed, all this can be done from the IoT server account without forgetting to disable the conditions given in table 3.2. The manual monitoring of the door is illustrated in figure 3.22.



Figure 3.22. Door manually monitoring.

3.6.3. Room

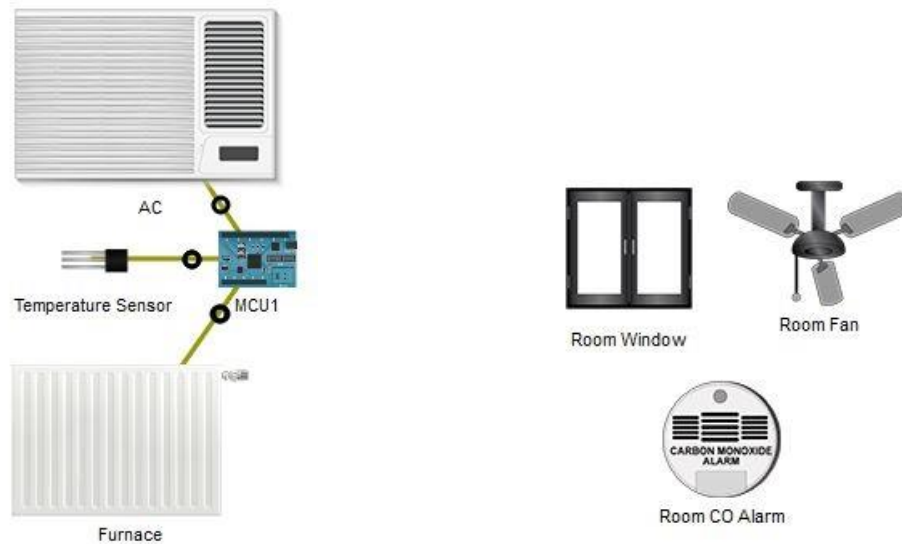
The room system consists of two parts: the first is designed to monitor the temperature and the second is to vent Carbon Monoxide to avoid suffocation.

3.6.3.1. Temperature monitoring design

The Furnace and the AC can be registered in the IoT server. We follow the same procedure to register them. Then, to perform the full automation here, we need a temperature sensor; it is available in Packet Tracer, and a microcontroller MCU1 (see figure 3.23(a)). We connect slot D0 and D1 of MCU1 to the AC and the Furnace respectively and slot A1 to the temperature sensor. The furnace in Packet Tracer increases the temperature of typical office space at 10C per hour. The AC Cools the temperature of typical office space at -10C per hour.

In this project, especially in summer, the AC is programmed to start cooling if the temperature exceeds 30°C, and to remain ON, until the temperature falls under 20°C, the AC then goes back OFF and remains OFF until the temperature exceeds 30°C again. In winter, the furnace is programmed to start heating the room if the temperature falls under 10°C and to remain ON until the temperature exceeds 20°C, the furnace then goes back OFF and remains OFF until the temperature falls under 10°C again.

Python is also used to program MCU1 microcontroller to perform this task; the program is given in Appendix B. The temperature sensor gives values from 0 to 1023 (mapping from -100C to 100 C). We can also switch the AC and the Furnace ON and OFF manually, using the IoT server account.



(a) Auto temperature monitoring design. (b) Auto CO monitoring design.

Figure 3.23. Room monitoring system design.

3.6.3.2. Vent Carbon Monoxide monitoring system

All of the devices on this part and which are shown in figure 3.23(b) can be registered in the IoT server. The CO Alarm is used as an alarm and a sensor at the same time, the alarm will go on if the Carbon monoxide in the room is more than 20%. The CO Alarm in packet tracer, by default, is programmed to go on when it detects a Carbon Monoxide level of 20%. To perform the automation of this part, the conditions given in table 3.4 must be set in the IoT server conditions TAB.

Table 3.4. Automatic Vent Carbon Monoxide monitoring system conditions.

Function name	Condition	Action
Room	Room CO Alarm level is between 20% and 25%	-Set Room Window On to true -Set Room Fan Status to Low
Room1	Room CO Alarm Level > 25%	-Set Room Window On to true -Set Room Fan Status to High
Room2	Room CO Alarm Level < 2%	-Set Room Window On to false -Set Room Fan Status to Off

The first condition is set to open the window and to start the Fan in its Low state when the CO level is between 20% and 25%. When the CO level exceeds 25%, the second condition is set to make the Fan in its High state and to keep the window open. The third condition works when the CO level is under 2%; it closes the window and turns the Fan off. If these conditions are disabled, the window can be opened manually using the IoT server account as illustrated in figure 3.24.

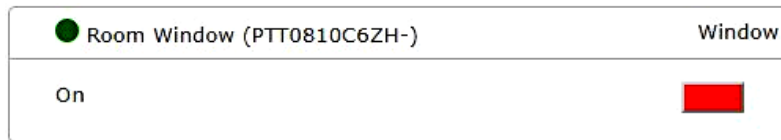


Figure 3.24. Manual Window monitoring IoT server account.

We can also switch between the three states of the Fan using the same TAB of the IoT server as shown in figure 3.25.

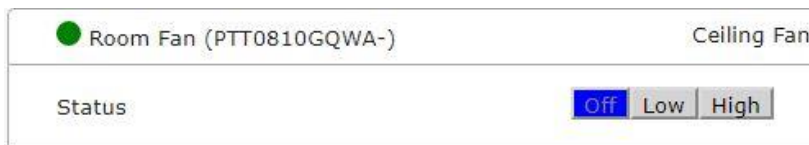


Figure 3.25. Manual Fan monitoring IoT server account.

Furthermore, we can see whether the alarm is on or off as shown in figure 3.26; also, the CO alarm provides the server by the CO percentage on the environment.

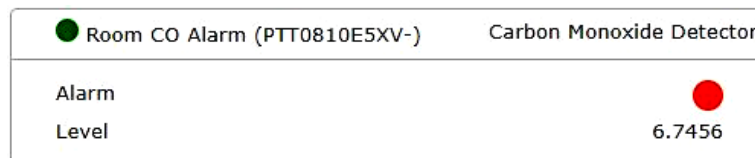


Figure 3.26. Manual Carbon Monoxide alarm monitoring IoT server account.

3.6.4. Garden

The garden in this project consists of three parts. The first part is responsible for the fire monitoring (see figure 3.27(a)). The second part is responsible for the automatic irrigation (see figure 3.27(b)), and the last part is for the CO level monitoring to avoid crop losses (see figure 3.27(c)).

3.6.4.1. Fire monitoring:

Both the fire alarm and the fire sprinkler are allowed to be registered in the IoT server. The fire alarm sends a “High” signal to the server whenever a Fire is close to it, or whenever it senses Fire.

The Fire simulation is not available in packet tracer. A thing can be created and programmed to have the IR property to act as a fire.

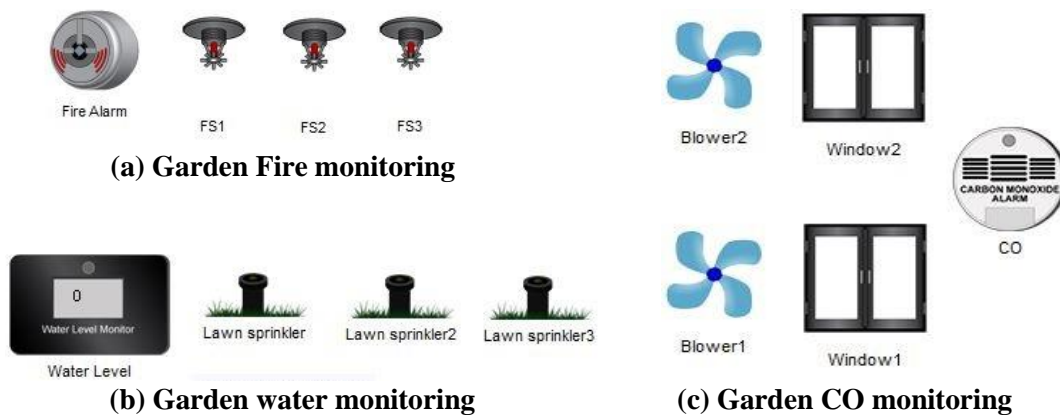


Figure 3.27. garden monitoring system.

We have to set conditions on the IoT server so that if there is fire, the server sets the fire sprinklers to ON. And if the alarm doesn’t detect any fire, the server sets the fire sprinklers to OFF. These conditions are summarized in Table 3.5. We can also use the IoT server account to turn ON and OFF the sprinklers manually, and see the fire alarm status.

Table 3.5. Garden fire monitoring system conditions.

Function name	Condition	Action
FS-ON	Fire Alarm Fire Detected is true	-Set FS1 Status to true -Set FS1 Status to true -Set FS1 Status to true
FS-OFF	Fire Alarm Fire Detected is false	-Set FS1 Status to false -Set FS1 Status to false -Set FS1 Status to false

3.6.4.2. Water monitor:

Both the water monitor level and the lawn sprinkler are allowed to be registered in the IoT server. The lawn sprinkler raises the water level each time it goes ON. The water level monitor gets the water level of the environment and prints it in “cm”. Some conditions in the IoT server, and summarized in table 3.6, should be added to automate this part.

The first condition sets the sprinklers OFF if the level measured by the water level monitor exceeds 10cm. The second condition turns the sprinklers ON if the water level goes less than 1cm. Of course, the garden will not be watered all the day, for that, the conditions can be disabled whenever we want. And we can make the watering manually from the IoT server account as shown in figure 3.28.

Table 3.6. Conditions in the IoT server to automate garden watering.

Function name	Condition	Action
LS-OFF	Water Level > 10.0cm	- Set Lawn sprinkler Status to false - Set Lawn sprinkler2 Status to false - Set Lawn sprinkler3 Status to false
LS-ON	Water Level < 1.0cm	- Set Lawn sprinkler Status to true - Set Lawn sprinkler2 Status to true - Set Lawn sprinkler3 Status to true

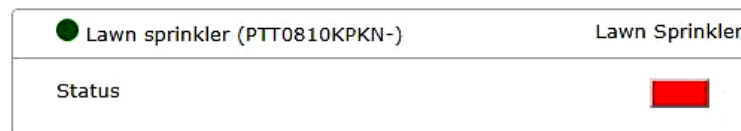


Figure 3.28. Enable and disable the watering operation manually.

3.6.4.3. Garden CO monitoring:

We have already known about the windows and the CO alarm; besides, the blower can be registered in the IoT server. There is no difference between the CO alarm that will be used here and the one that is used in the room.

The garden is bigger than the room; so, two windows are used and a blower is placed near each window. Some conditions are set in the IoT server to achieve the automation. These conditions are summarized in table 3.7.

Table 3.7. Conditions in the IoT server to automate the garden CO monitoring.

Function name	Condition	Action
CO-ON	CO Alarm is true	- Set Window1 On to true - Set Window2 On to true - Set Blower1 Status to High - Set Blower1 Status to High
CO-OFF	CO Alarm is false	- Set Window1 On to false - Set Window2 On to false - Set Blower1 Status to OFF - Set Blower1 Status to OFF

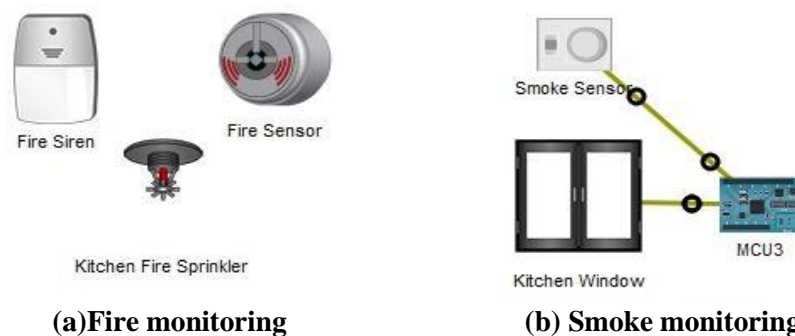
If the CO level in the environment exceeds 20%, the alarm goes ON. The level of the CO is not controlled, only the alarm is used to decide whether to open the windows and start the blowers or not. For this part also, the conditions can be disabled, to access the manual control of the windows and the blowers.

3.6.5. Kitchen

The kitchen consists of two parts: one for fire monitoring, and the other to vent Smoke to avoid suffocation.

3.6.5.1. Kitchen fire monitoring:

The fire alarm and the fire sprinkler were discussed in the Garden part; for the kitchen a siren is added to notify people who are inside the house, it can be registered in the IoT server. Same design as that is used for the garden is used for this case; however, unlike the garden, one sprinkler is enough as shown in figure 3.29.

**Figure 3.29. Automatic Kitchen Fire and smoke monitoring.**

The conditions that must be added are: (i) a condition to start the fire sprinkler and the siren if the fire alarm detects fire in the kitchen, and (ii) a condition to stop the sprinkler and the siren if there is no fire in the kitchen. Table 3.8 summarizes these two conditions.

Table 3.8. Kitchen Fire and smoke monitoring conditions.

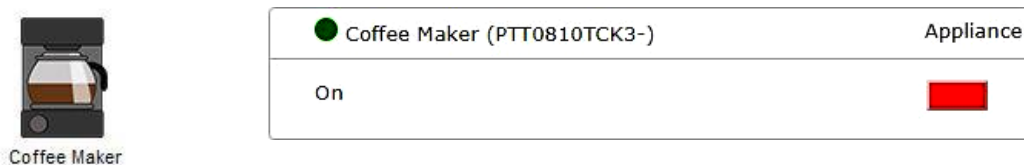
Function name	Condition	Action
SirenON	Fire Sensor Fire Detected is true	-Set Kitchen Fire Sprinkler Status to true -Set Fire Siren On to true
SirenOff	Fire Sensor Fire Detected is false	-Set Kitchen Fire Sprinkler Status to false -Set Fire Siren On to false

3.6.5.2. Vent Smoke to avoid suffocation:

For this part there is only the window that can be registered in the server. Smoke sensor is not a smart component. It measures the smoke level in the environment and sends analog output in range of 0 to 255, representing 0 to 100% smoke percentage in the air using an IoT cable. So, the microcontroller MCU3 is needed to achieve the full automation to vent smoke. First, slot A0 of MCU3 is connected to the smoke sensor and A1 to the window.

Python is used to program MCU3 (see Appendix C). The sensor provides the level of the smoke in the environment. MCU3 gets new values every second. If the smoke level is more than 20%, the window gets digital “1” from MCU3, it opens. If the smoke level is less than 2%, the window gets digital “0” from MCU3, and it closes. Otherwise, the window stays in its previous state, waiting for a new decision.

Since the window is registered in the IoT server; so, it can be opened or closed manually. Furthermore, packet tracer has another IoT device, which is a coffee machine; it can be registered in the server. When we access the IoT server account we can turn it on to prepare a coffee as shown in figure 3.30.

**Figure 3.30. Coffee machine enabling.**

3.7. Conclusion

In the end of this chapter, All the steps were mentioned and explained so that the IoT-based smart home will be locally monitored using a laptop inside the home, and remotely monitored using a 3G/4G client outside the home through the Internet.

After any configuration, we need to test all the devices whether they are well configured or not, and if they act successfully under some conditions and change in the environment. In the next chapter, we will try all the possibilities for each device and observe the behavior.

4

TESTS AND RESULTS

4.1. Testing the manual usage of the IoT devices

For this part, we will access the IoT server account that was configured before through the 3G/4G client that is connected to the cell tower of the “INELEC” provider.

By typing the domain name “myhome.org” in a web browser, we are testing both the **DNS server** and the **IoT server**, if the domain name is translated to the IoT server IP address “192.168.1.3” successfully, that means the DNS is working, and if we can access the IoT server, and the two fields (username and password) have shown up, that means the IoT server is working. Figure 4.1 shows the web browser of the client when we type the URL (Uniform Resource Locator) “myhome.org”

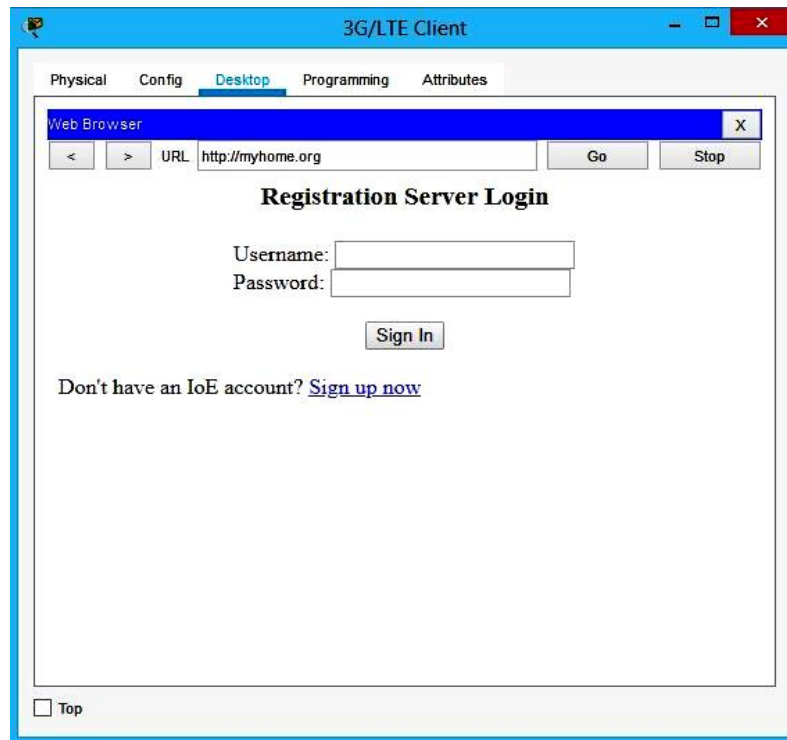


Figure 4.1. The IoT server interface from the 3G/4G client web browser.

Then, the authentication credentials (username and password) that were configured before are used to sign in the account.



Figure 4.2. the authentication credential for the IoT account.

After signing in, the IoT devices that are registered before will show up.

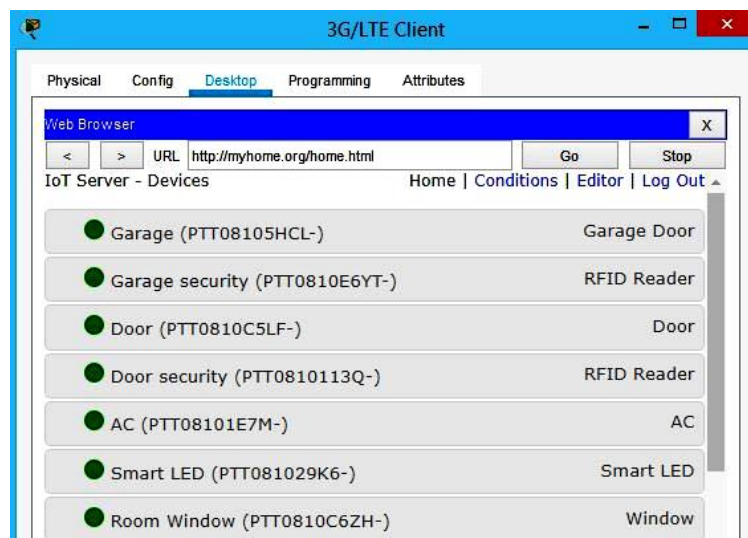


Figure 4.3. Home interface of the IoT account.

Finally, we can change the states of most of the devices remotely from the 3G/4G client's web browser without forgetting to disable the conditions.

4.1.1. The garage

The garage, by default, is closed as shown in figure 4.4.

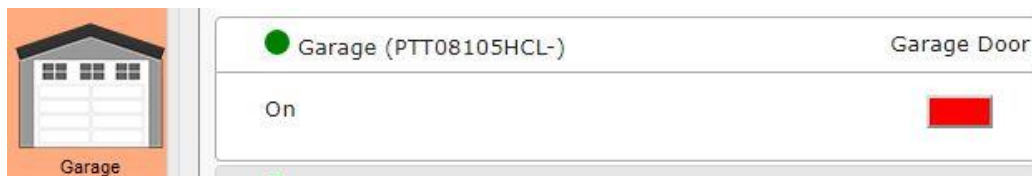


Figure 4.4. Closed garage.

After clicking on the red rectangle, it turns green as illustrated in figure 4.5 and opens the garage.



Figure 4.5. Open garage.

4.1.2. The door

The door also, by default, is locked.

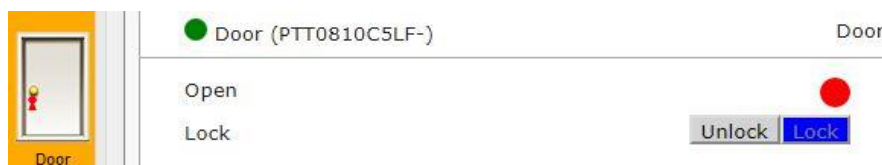


Figure 4.6. Locked door.

After clicking the "unlock" bottom, it unlocks the door. The door is opened manually by clicking the door itself (ALT+click).



Figure 4.7. Unlocked door.

4.1.3. The windows

We have four windows, one in the room, one in the kitchen, and two in the garden. The windows are, by default, are closed as shown in figure 4.8. After clicking on the red rectangle, it turns green and opens the window as illustrated in figure 4.9. And this is valid for all the windows.

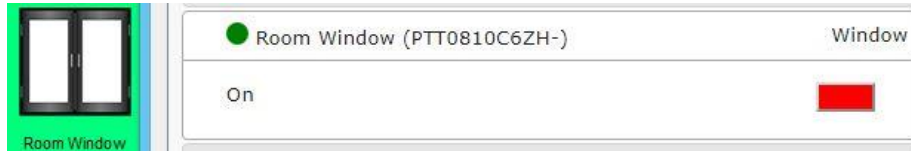


Figure 4.8. Closed window.



Figure 4.9. Open window.

4.1.4. The room FAN

The room FAN has three states, OFF, Low, and High. After disabling the conditions that are associated with the FAN, we can switch between the three states.



Figure 4.10. The FAN status is OFF.

If we click on “Low”, the fan starts working at low speed as shown in figure 4.11. Then, if we click on “High”, the fan increases its speed as illustrated in figure 4.12.



Figure 4.11. The FAN status is LOW.



Figure 4.12. The FAN status is HIGH.

4.1.5. The AC

The AC's automation is performed by a microcontroller, not the IoT server, so there are no conditions to be disabled. It can be set ON/OFF manually. After clicking on the red rectangle shown in figure 4.13, it turns green and AC is turned ON. One can see a small red led in the bottom-right of the AC if the AC is on (see figure 4.14).



Figure 4.13. The AC is turned OFF.



Figure 4.14. The AC is turned ON.

4.1.6. The furnace

The same thing for the furnace automation, it can be set ON/OFF manually. Then, after clicking on the red rectangle, it turns green and the furnace is turned ON. For the furnace, the red led that indicates that the furnace is ON is placed in the top-right of it see figure 4.16.



Figure 4.15. The furnace is turned OFF.

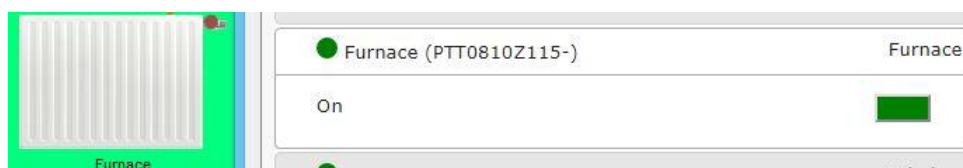


Figure 4.16. The furnace is turned ON.

4.1.7. The fire sprinklers

After disabling the conditions that are assigned to the fire sprinklers, either the sprinkler of the kitchen or the three sprinklers of the garden can be accessed manually as shown in figure 4.17. Then, if we click on the red rectangle, the rectangle will be green and the fire sprinkler starts sprinkling water as shown in figure 4.18.



Figure 4.17. The fire sprinkler is OFF.

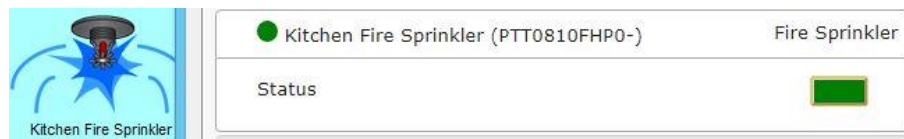


Figure 4.18. The fire sprinkler is ON.

4.1.8. The lawn sprinklers

After disabling the conditions that are assigned to the fire sprinklers, they can be started and stopped manually. By clicking on the red rectangle, the rectangle will be green and the lawn sprinkler starts sprinkling water as illustrated in figure 4.20.



Figure 4.19. The lawn sprinkler is OFF.



Figure 4.20. The lawn sprinkler is ON.

4.1.9. The Blowers:

The blower has three states, OFF, low, and high. After disabling the conditions that are associated with the two blowers of the garden, we can switch between the three states. If we click on “Low”, the blower starts working at low speed. Then, if we click on “High”, the blower increases its speed as illustrated in figures 4.21, 4.22 and 4.23.

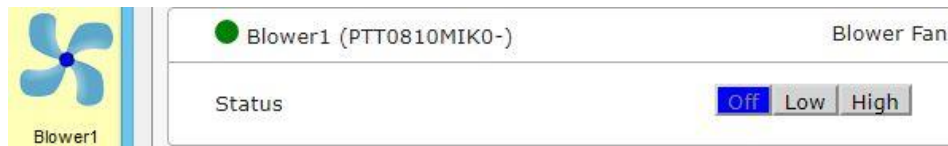


Figure 4.21. The blower status is OFF.

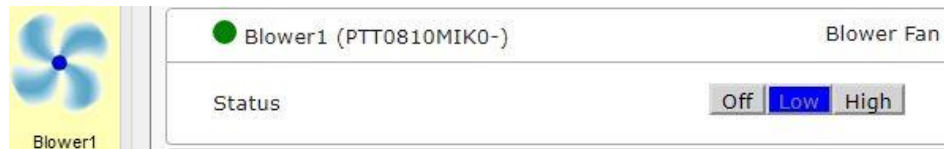


Figure 4.22. The blower status is LOW.

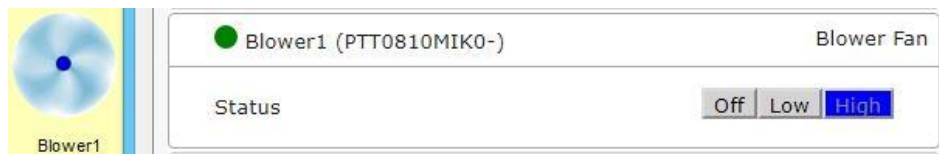


Figure 4.23. The blower status is HIGH.

4.1.10. The siren

First, we need to disable the conditions that turn the siren ON and OFF; then, we will be able to turn it ON/OFF manually. The same for this device, a click on the red rectangle will turn it green and turn the siren ON, the white part of the siren will be red if it is ON.



Figure 4.24. The siren is OFF.



Figure 4.25. The siren is ON.

4.1.11. Coffee maker

The coffee maker is an example of a simple appliance, it can, for sure, be manually turned ON and OFF. Then, like the others, a click on the red rectangle will turn it green and start the appliance to make coffee, a red led will appear in the bottom-left of the appliance if it is ON (see figures 4.26 and 4.27).



Figure 4.26. The coffee maker is OFF.



Figure 4.27. The coffee maker is ON.

Those are the IoT devices that can be monitored manually from the IoT server, some other devices can be only supervised, such as, smart LEDs, RFID readers, CO alarms, and the water level monitor.

Another important feature in the Cisco Packet Tracer tool is the possibility to separate the network at a physical level using sub-environments such as city, building, containers, and wiring cabinets. From the main view of the tool, it is possible to switch very quickly between the logical and physical layers as illustrated in Figure 4.28.



Figure 4.28. The logical and physical view.

When creating network simulations, in the logical view, by default all the components are placed in the same physical space. For basic simulation this is not probably a detail that should be taken care of; however, for IoT simulations, it was advisable to utilize different physical layers in order to adjust the environment variable to influence the IoT devices' behavior.

The presentation of this work will not be a real-time presentation, the devices' behavior will be presented one by one, so there is no need to separate them, the environment variable can be changed whenever we want.

4.2. Environment variables

Every physical sub-layer, except for the wiring cabinets, are coming with a large set of fully customizable environmental variables.

Variables are tunable parameters for representing real life environments such as the amount of sunlight, temperature, air carbon dioxide and monoxide concentration, water level, and many more. In the Cisco Packet Tracer, there are more than fifty different variables that can be adjusted accordingly based on a 24h time range. In Figure 4.29 below one can see the amount of sunlight and smoke during the whole day.

Variables are necessary to influence the sensor behavior in the IoT simulations. Variables are detected by the sensor and, as a consequence, actions are then triggered. Adjustable variables were also helping us to validate immediately the IoT logic setup.

In the provided simulation the variables utilized were: (i) sunlight to manage the lighting of the door and the garage, (ii) Carbon monoxide level to be aware of the CO level in the room and the garden so that the windows will be opened automatically (for this part an old car can be also used to generate CO in the environment), (iii) the ambient temperature level was used during a random time of the day in order to activate the temperature sensor in the room and (iv) the water level was used by the water lever monitor in the garden to detect extra water levels and stop the water irrigation sprinklers.

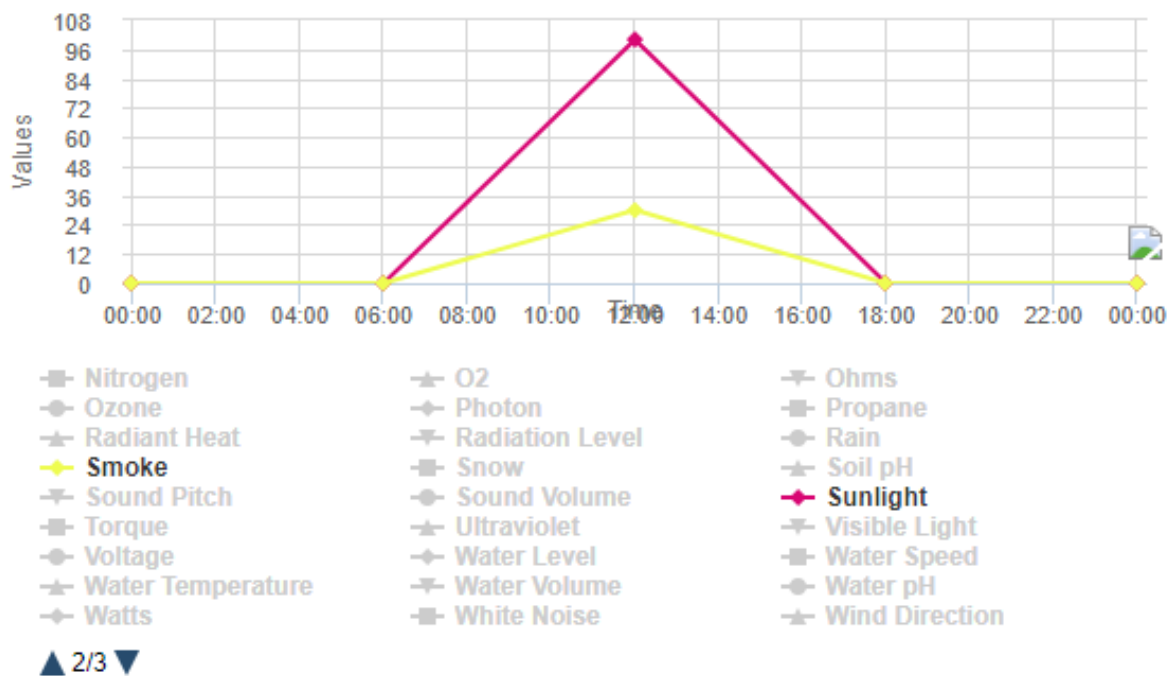


Figure 4.29. Example of environmental variable setup.

4.3. Testing the automation of the IoT devices

4.3.1. The garage

For this part, we will test the behavior of the garage using the two cards (My car) and (Not my car), and the smart LED whether there is the sunlight or not. Hence, we need to set up values for the sunlight variable as illustrated in figure 4.30.

It is clearly shown in figure 4.30 that from 6 PM to 6 AM the sunlight variable is 0. Whereas, from 6 AM to 6 PM (during the day time) there will be a change in the amount of the sunlight.

To validate our results, three scenarios are created as explained below:

Scenario 1: We bring the authorized card (My car) near the RFID reader and check what happens to the garage. At the same time, we make sure that there is no sunlight by setting the time in the range of 6 PM to 6 AM. As there is a car near the garage; thus, there will be a motion near the motion detector. Hence, the smart LED light-up for 35 seconds, and the garage opens for 30 seconds.

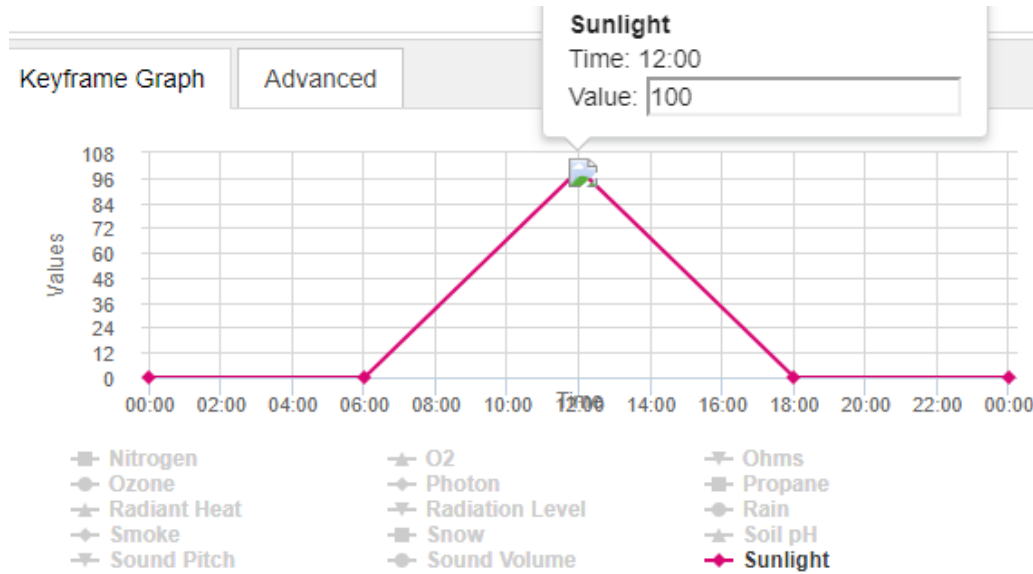


Figure 4.30. The change of the Sunlight (percentage) over 24 hours.

The result is as expected, the garage is open for 30 seconds, and then it goes back close. The smart LED also stays ON for 35 seconds and goes back OFF. So, the first scenario was validated (see figure 4.31).

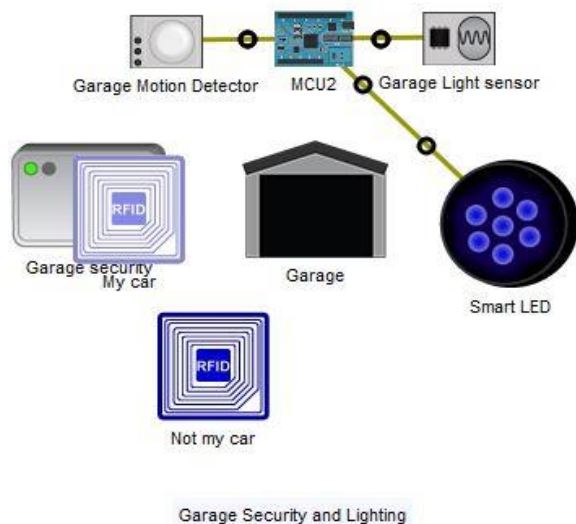


Figure 4.31. An authorized card and motion while there is no sunlight

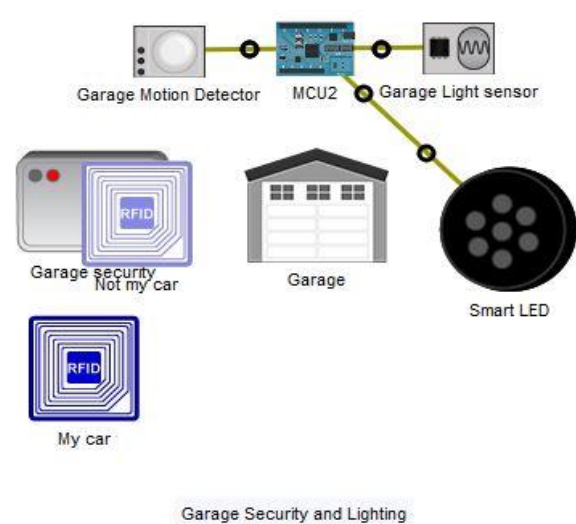


Figure 4.32. An unauthorized card and motion while there is sunlight

Scenario 2: We bring the unauthorized card near the RFID reader and check what happens to the garage. At the same time, we make sure that there is sunlight by setting the time in the range 6 AM to 6 PM, and make a motion near the motion sensor. Hence, the smart LED won't light up, and the garage won't open. The result also is as expected, the garage is still closed and the smart LED is still OFF. So, the second test was successful (see figure 4.32).

Scenario 3: Do not make any card near the RFID reader, and choose a time in the interval (6 PM to 6 AM) as shown in figure 4.33, so that the sunlight is 0. Therefore, the RFID reader is in the “waiting” state, so the garage won't open, and despite the absence of the sunlight, there is no motion the smart LED won't light up as shown in figure 4.34.

Current Time: 01:00:00

Figure 4.33. Adjusting the time to 1 AM.

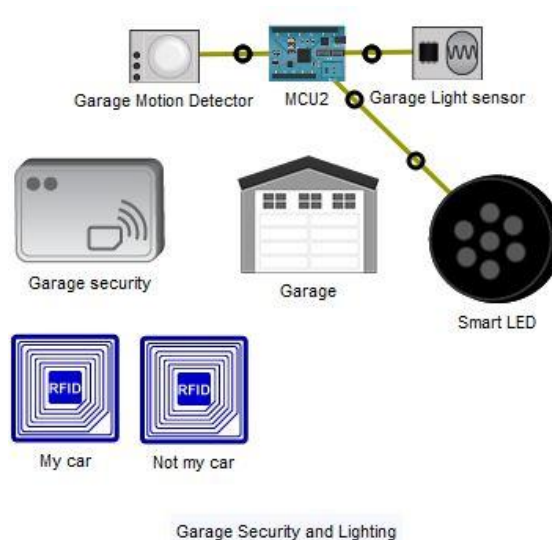


Figure 4.34. Garage security in “Waiting” state and no motion while there is no sunlight.

4.3.2. The door

For this part, the security of the door is tested using the two cards (A family member) and (A stranger). The same values for the sunlight variable that are used to test the garage lighting are used for this case, and three scenarios are created as follow:

Scenario 1: The authorized card (of family member) is brought close to the RFID reader to check what happens to the door. At the same time, we will make sure that there is sunlight by setting the time in the range 6 AM to 6 PM and ensure motion near the motion sensor (if there is someone near the door, the motion detector will be activated). In this case, the door is unlocked for 10seconds so

that the person can enter the home and then the door will be locked again, and the smart LED is still OFF because the light sensor detected sunlight. Hence, the first test is successful (see figure 4.35).

Scenario 2: in this case unauthorized card (of stranger) is brought near the RFID reader to check what happens to the door. At the same time, we make sure that there is no sunlight by setting the time in the range of 6 PM to 6 AM, and ensure motion near the motion sensor. Figure 4.36 illustrates that the door is still unlocked, and the smart LED is ON for 15 seconds, and then it goes OFF again. Thus, means that the second scenario is validated.

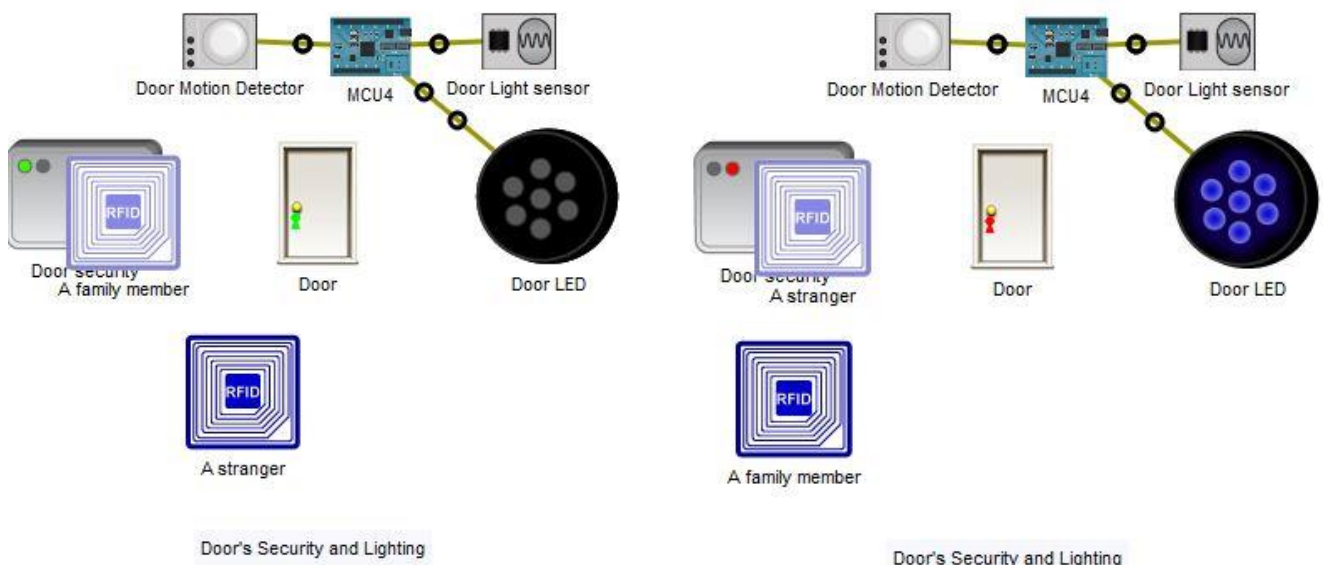


Figure 4.35. An authorized card and motion while there is sunlight

Figure 4.36. An unauthorized card and motion while there is no sunlight

Scenario 3: for this case no card exists near the RFID reader, and the time is set in the interval of (6 PM to 6 AM), so that the sunlight is 0 as shown in figure 4.37.

Current Time: 05:30:00

Figure 4.37. Adjust the time to 5:30 AM.

Both the RFID readers of the garage and the door, if they are in the “waiting” state the garage won’t open and the door won’t be unlocked, and despite of the value of the sunlight since there is no motion, the two smart LEDs won’t light up. So, the third test of the garage and the third test of the door were successful (see figure 4.38).

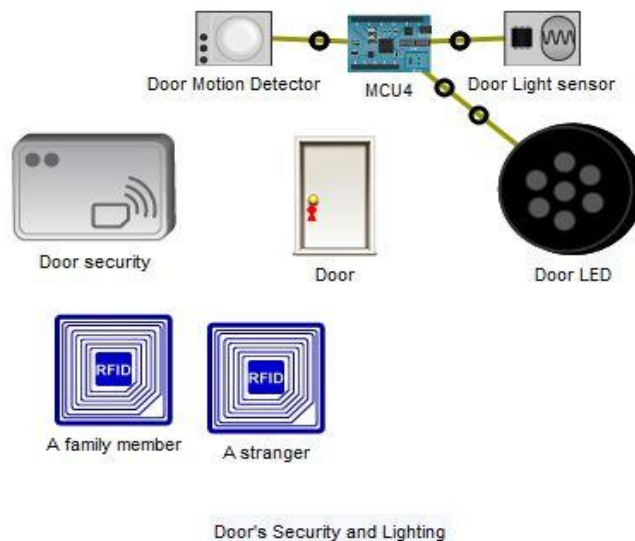


Figure 4.38. Door security in “Waiting” state and no motion while there is no sunlight.

4.3.3. The automatic CO monitoring

There are two ways to control the level of CO in the environment: even by changing its values from the environment TAB like for the sunlight, or by using an old car (once it is ON CO level increases). The two ways are available in Packet Tracer.

There are two CO monitoring systems in this home: Room CO monitoring system and garden CO monitoring system each one of them is validated by two scenarios.

a) Room CO monitoring system

Scenario 1: An old car is inserted and turned on (it can be activated by the shortcut ALT+click); thus, the CO level increases. The CO level can be supervised by the IoT server. Normally, when the CO level exceeds 20%, the window opens and the fan runs in its Low speed.

Figure 4.39 illustrates that the first test is successful: when the CO level exceeds 20% the red led is turned on, the alarm appears, the room window opens and the fan runs in its low speed to vent CO.

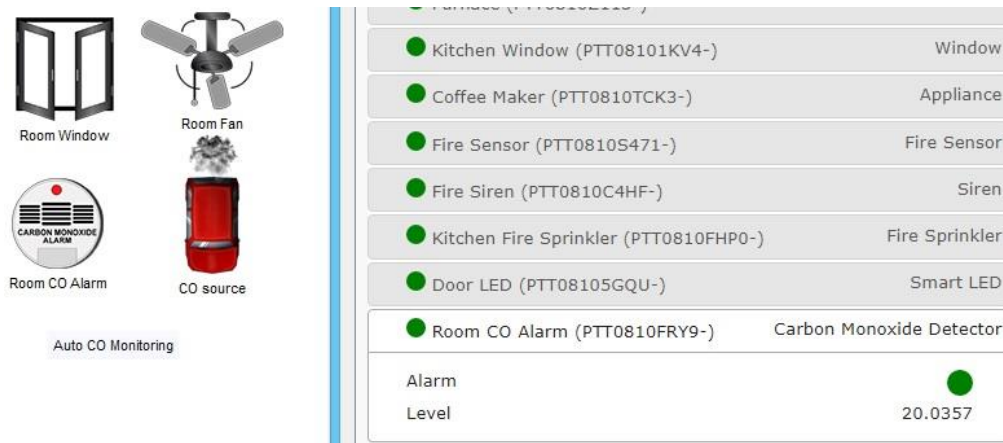


Figure 4.39. The room window is open and the room fan is working once the CO level exceeded 20%.

Scenario 2: The window and the fan are not sufficient to vent the entire CO from the room as the CO level is still increasing, so they will stay on until the car is turned off. Once, the CO level falls under 2%, the window closes and the fan turns OFF. So, this test is successful validated.



Figure 4.40. The room window is closed and the room fan is OFF after the CO level is less than 2%.

b) Garden CO monitoring

Scenario 1: The same procedure is used for the garden system. If the CO level exceeds 20%, the two windows will be opened and the two blowers will run in their high speed. Figure 4.41 shows that the first test is successful, when the CO level exceeds 20% the red led on the alarm appears, the garden windows open and the blowers run in their high speed to vent CO.

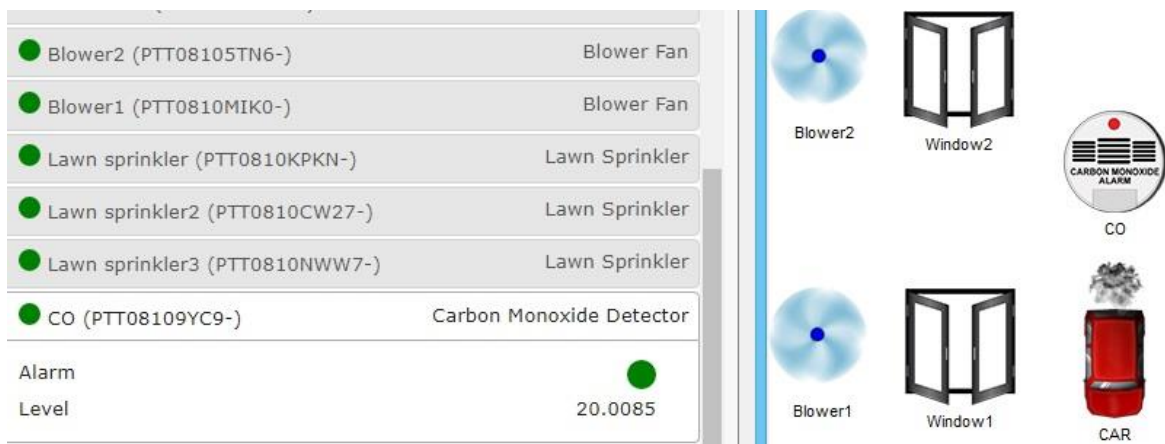


Figure 4.41. The garden windows are open and the blowers are working after the CO level exceeded 20%.

Moreover, as the CO level still increasing; the windows and the blowers will not be able to vent the entire CO from the garden unless the CAR is stopped from producing the CO. At this situation, the CO level falls under 2%, the two windows will be closed and the two blowers will be turned OFF. So, this test is also successful as illustrated in figure 4.42.

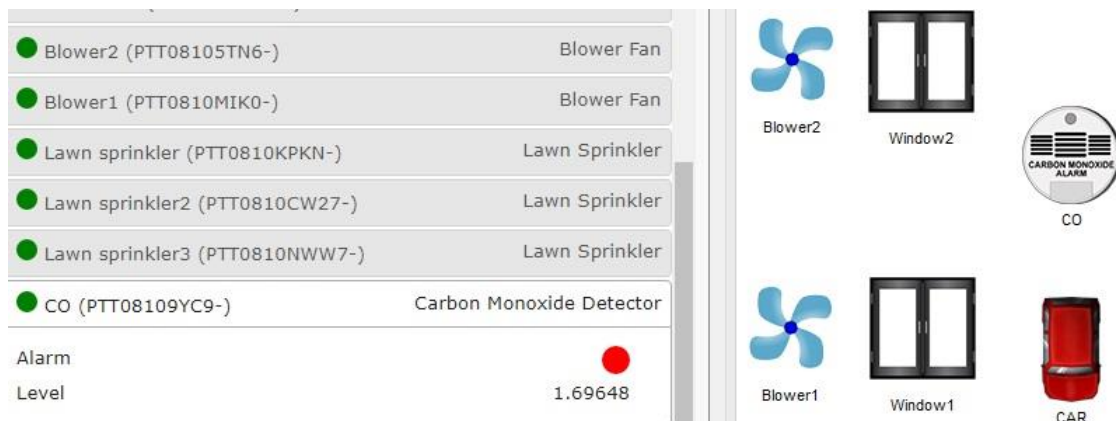


Figure 4.42. The garden windows are closed and the blowers are OFF after the CO level is less than 2%.

4.3.4. The automatic temperature monitoring

It is composed of two systems: room cooling system and room heating system. So, appropriate values must be set for the ambient temperature, and both of them can be tested by changing the current time. For the AC, the temperature must be set in a range of 15°C and 30°C. Whereas, for the furnace the temperature must be set in a range of 0°C and 15°C. Figure 4.43 summarizes the ambient temperature variation used in our test during 24 hours.

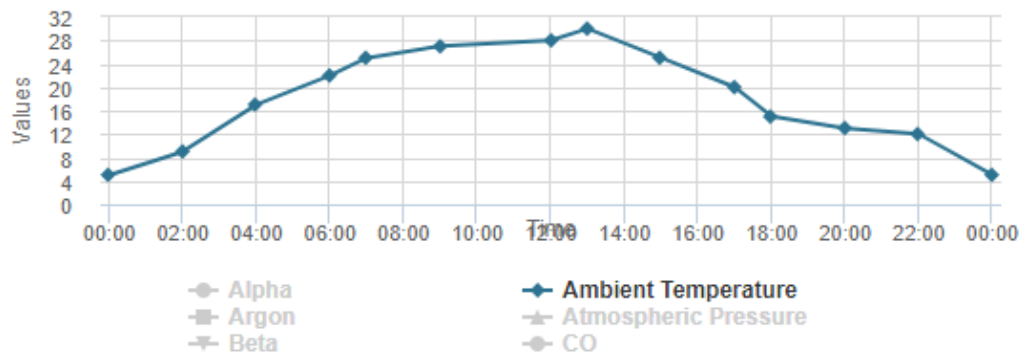


Figure 4.43. The change of the ambient temperature for 24 hours.

a) The Furnace system

First, the performance of the furnace is tested if the temperature is less than 10°C ; thus, the time is adjusted around midnight. Temperature monitor is used to show the current temperature. Therefore, as temperature changes versus time we observe the furnace functioning.

The furnace is configured to start running if the temperature is less than 10°C and keeps heating until the temperature reaches 20°C , and that what is illustrated in Figure 4.44. Thus, the test is successful.

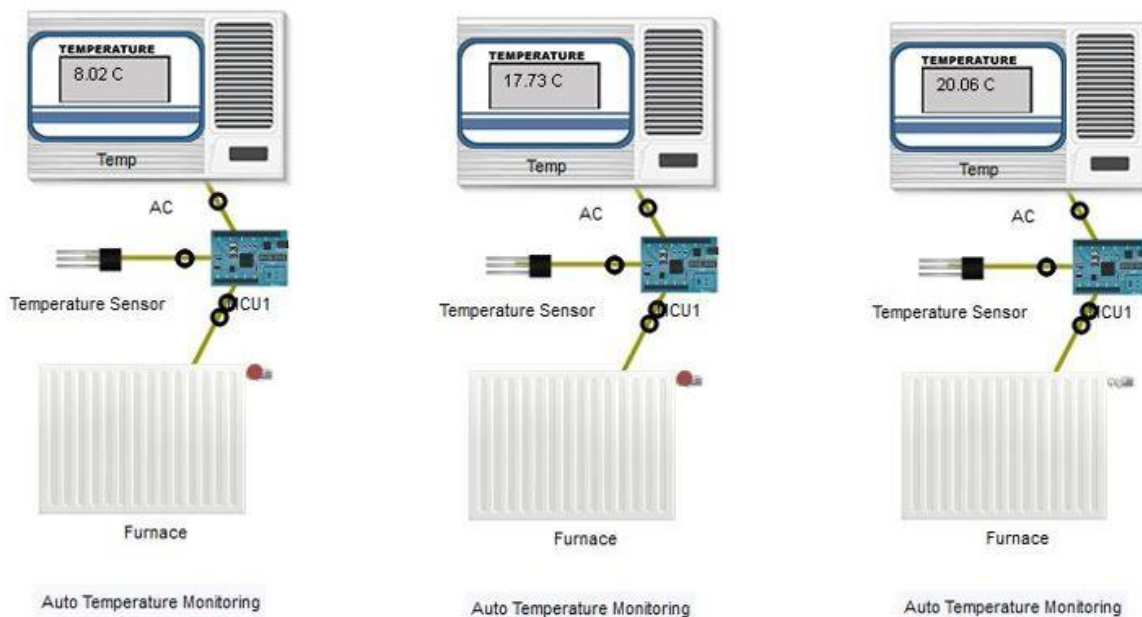


Figure 4.44. The status of the furnace versus temperature variation.

Second, the furnace is tested after it stops heating and when the temperature falls under 10°C again. Figure 4.45 below shows that if the temperature falls under 10°C , the furnace starts heating again which proves that the decreasing temperature test is successful.

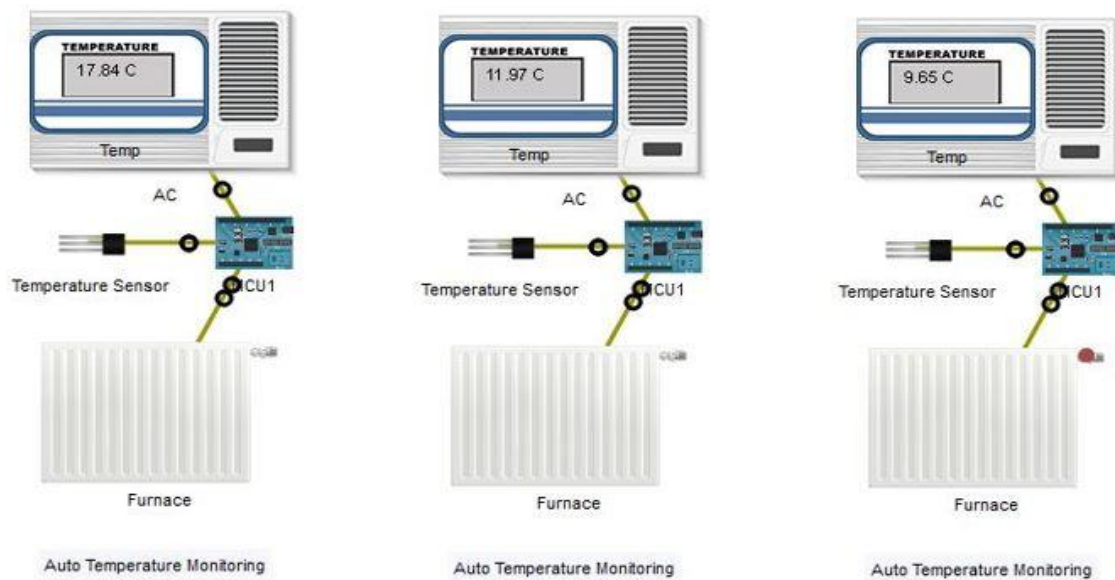


Figure 4.45. The status of the furnace versus the temperature decreasing.

b) The AC

Approximately, the same test as done with the furnace is repeated here. Just, the time interval is changed to be around noon. The AC is configured to start functioning once the temperature exceeds 25°C and to keep cooling until the temperature falls less than 20°C. Figure 4.46 illustrates this operation. So, the decreasing temperature test is successful.

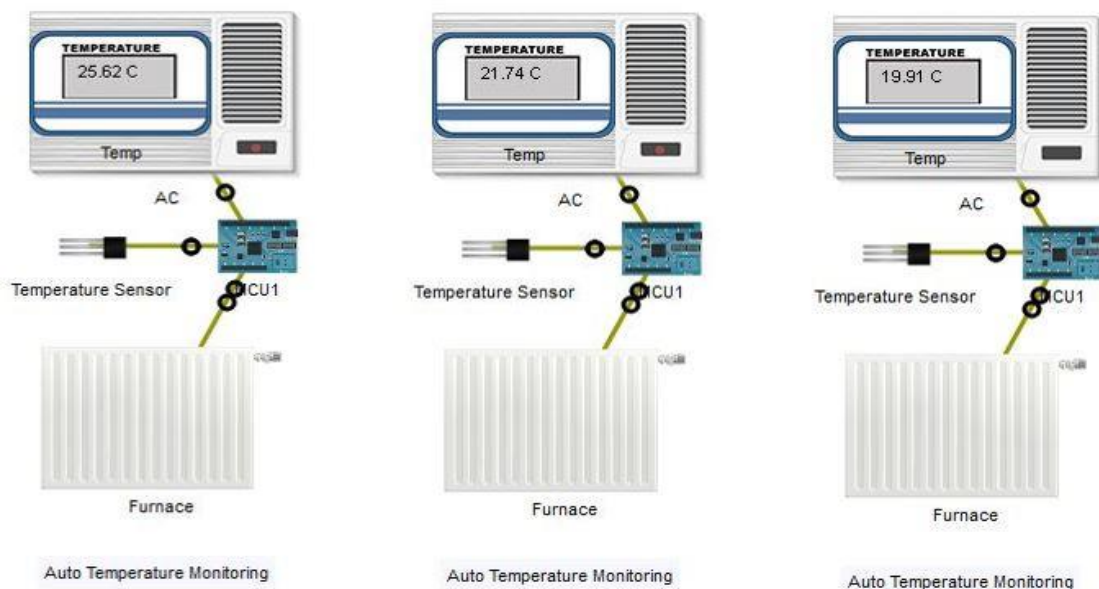


Figure 4.46. The status of the AC versus temperature decrease.

Figure 4.47 below demonstrates that once the AC stops cooling, it starts again when the temperature exceeds 25°C. The increasing temperature test is successful.

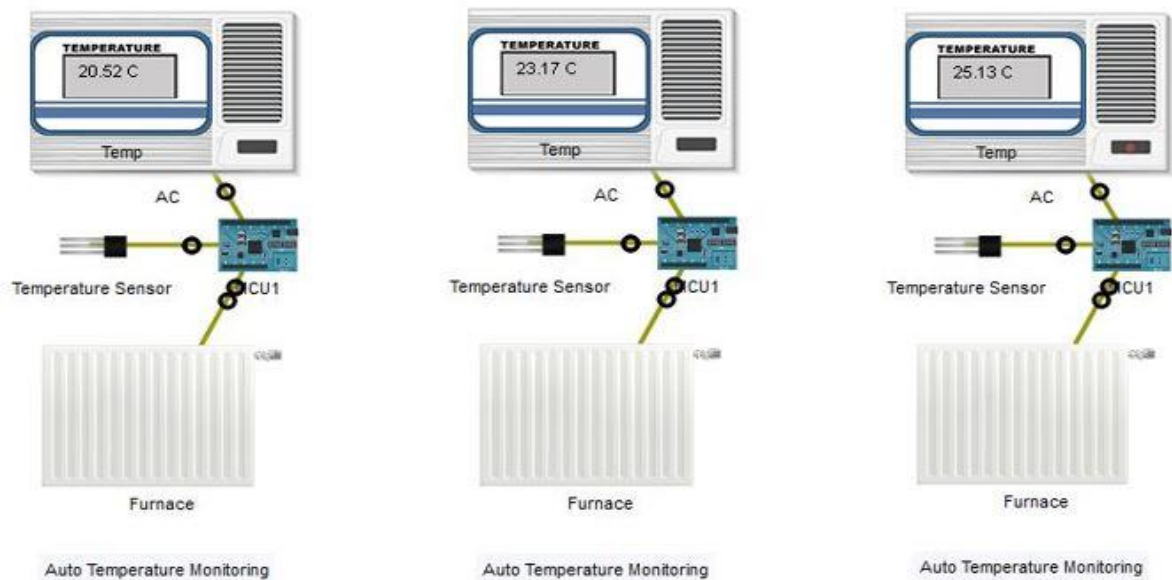


Figure 4.47. The status of the AC with increasing temperature at several moments.

4.3.5. The automatic smoke monitoring

Smoke is also an environment variable, there are two ways to test the performance of the smoke monitoring system: even by setting the values for the smoke variable from the environment TAB, or we by using an old car which generates smoke also. In this test, an old car is used because it is easier, permanent and not a real-time test.

First, smoke is generated using the old car (CAR 1) and wait until the smoke level reaches 20%. The smoke level can be checked from the attribute TAB of the smoke sensor.

Figure 4.48 illustrates that once the smoke level reaches 20%, the window opens to vent smoke from the kitchen. So, the first test is successful.

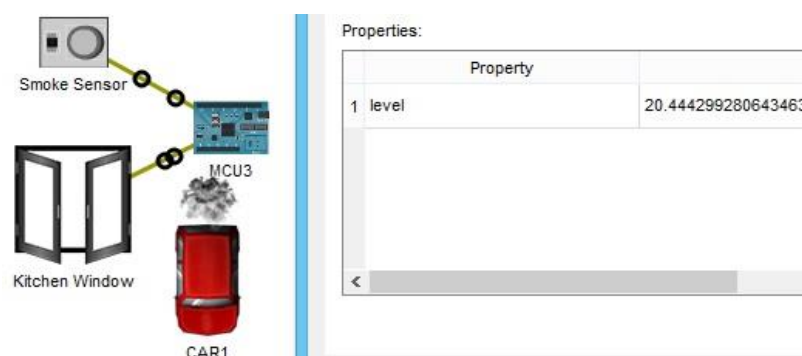


Figure 4.48. The kitchen window state after the smoke level exceeds 20%.

The window, by default, in Packet Tracer won't vent all the smoke from the kitchen, the level won't fall under 9%, so we need to stop the old car from increasing the smoke in the kitchen, and then see what happens when the smoke level falls under 2%.

Figure 4.49 below shows that the kitchen window closes when the smoke level falls under 2%, and this means that the second test is also successful.

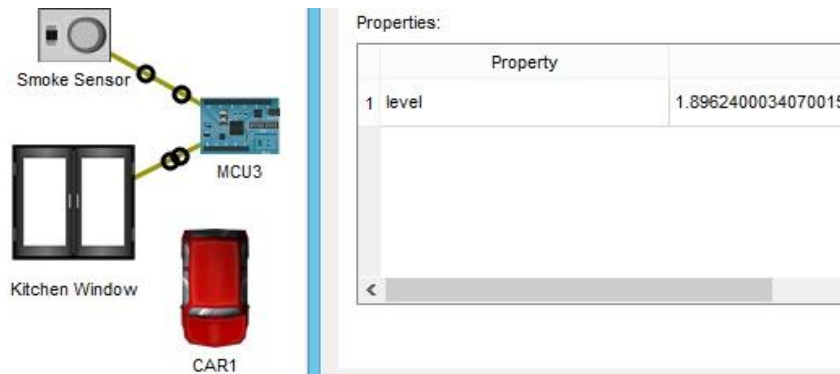


Figure 4.49. The kitchen window state after the smoke level falls under 2%.

4.3.6. The fire monitoring system

Fire is not included in the environment variables. So, objects that simulate fire are needed. These objects should have "IR" property with a value that can be considered as fire. See appendix D.

a) Kitchen fire monitoring

The siren and the fire sprinkler are tested in case there is fire and in the reverse case. Figures 4.50 and 4.51 show that if fire is detected in the kitchen, the siren goes ON and the fire sprinkler starts sprinkling water. So, the kitchen fire monitoring system is working.



Figure 4.50. The siren and the kitchen sprinkler if there is no fire.



Figure 4.51. The siren and the kitchen sprinkler in case of fire.

b) Garden fire monitoring

The same tests are repeated to test if there is or there is no fire in the garden. Figures 4.52 and 4.53 show that if fire is detected in the garden the fire sprinklers start sprinkling water. Hence, the garden fire monitoring system is verified.



Figure 4.52. The garden sprinklers if there is no fire.



Figure 4.53. The garden sprinklers if there is fire.

4.3.7. Garden water monitoring

The lawn sprinklers are configured to start sprinkling water if its level is bellow 1cm and stop when it exceeds 10cm. The water level monitor shows the level of water in the environment.

First, we start observing the behavior of the sprinklers if the water level is less than 1cm until it reaches 10cm. Figure 4.54 illustrates that the sprinklers kept sprinkling water until its level reached 10cm then they turn OFF. The test is successful.

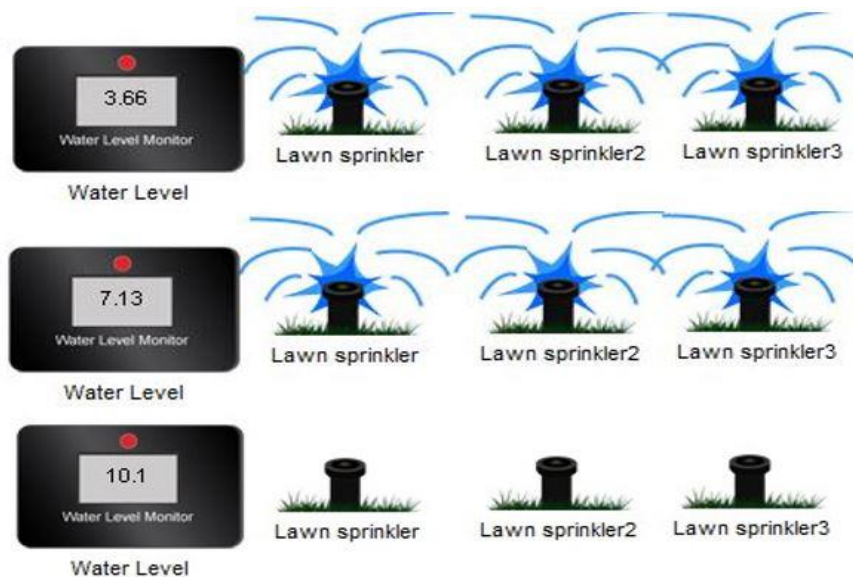


Figure 4.54. The behavior of the lawn sprinklers at three different moments.

The behavior of the sprinklers has been observed after the water level reached 10cm and when it decreases until it falls under 1cm.

The sprinklers are turned OFF when the water level exceeded 10cm and they stayed OFF until the water level fell under 1cm then they started sprinkling again. This scenario is illustrated in figure 4.55 proving that the test is successful.

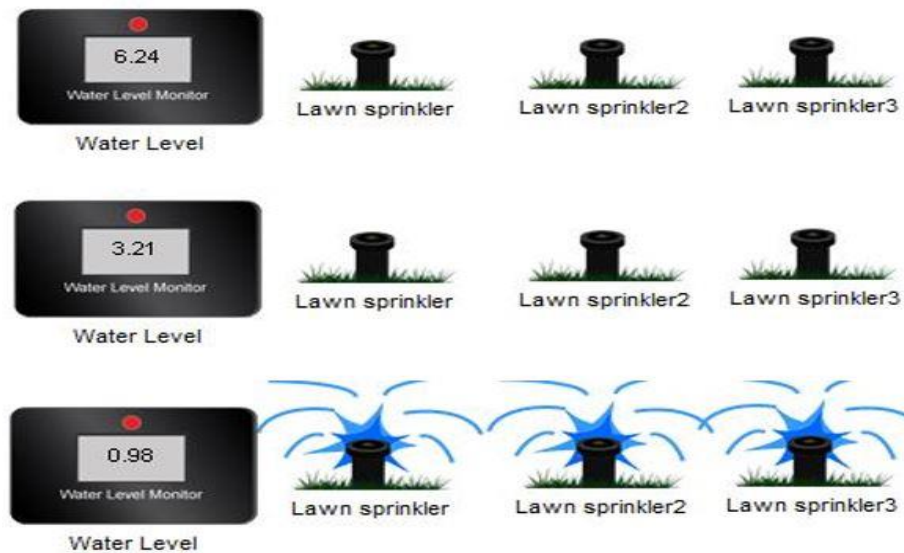


Figure 4.55. The behavior of the lawn sprinklers at another three different moments.

The process will keep repeating itself unless we disable the condition.

4.4. Conclusion

At this point, all the IoT devices that have been registered in the IoT server are working successfully either in their manual or the automatic usage. Some of the devices were unable to be registered in the IoT server that is why we used many microcontrollers; their tests were also successful.

Since we are reporting our project, we were unable to use real-time test, we couldn't report setting real values for the environment and observing the change every moment.

GENERAL CONCLUSION

As discussed in the fourth chapter, the overall conclusion on the project was appropriate, as the IoT simulations were delivered. Conclusions below cover both the deliverables and the methodology of the work.

The first goal was to apply our knowledge in networks and IoT to design a smart home based on IoT devices, and set-up the internet to have the ability to access the home remotely using a 3G/4G client. We successfully reached our goal using the CISCO packet tracer simulation tool, and the home was effectively designed and simulated.

Dividing the work into phases also helped to track down properly the progress. Starting by setting-up the internet provider and service provider (IoT and DNS), then connecting devices to the internet and registering them to the server. Finally, setting the conditions and the methods they will follow.

The Cisco Packet Tracer simulation should be treated as a starting point for future and more complex IoT simulations as Cisco Packet Tracer tool will be updated with more IoT functionalities. In addition, it is a very helpful tool for our future life as engineers dealing with real devices and smart homes.

To sum up, for future implementations of the project, changes in the agenda are advised, along with studies to explore different IoT simulator technologies and feasibility to utilize real IoT hardware.

APPENDIX A

Figure A.1 is a flowchart for the garage lighting process:

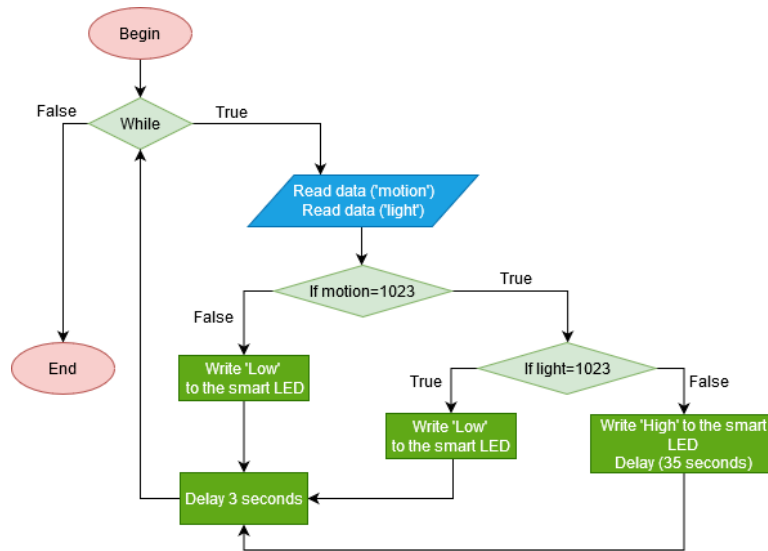


Figure A.1. Garage lighting flowchart.

Figure A.2 illustrates the Garage lighting python code:

```
1 from gpio import *
2 from time import *
3
4 def main():
5     pinMode(0, IN) #Light sensor pin
6     pinMode(1, IN) #Motion sensor pin
7     pinMode(2, OUT) #Smart LED pin
8     print("Getting data...")
9     while True:
10        motion=analogRead(1);
11        light=analogRead(0);
12        if (motion==1023):
13            if (light==1023):
14                analogWrite(2, [0]) # [0] --> "Low"
15            else:
16                analogWrite(2, [1023]) # [1023] --> "High"
17                delay(35000)
18        else:
19            analogWrite(2, [0])
20            delay(3000);
```

Figure A.2. Garage lighting python code.

The only difference for the door lighting is the ON time, 35 seconds will be only 15 seconds

APPENDIX B

Figure B.1 is a flowchart for the room temperature monitoring:

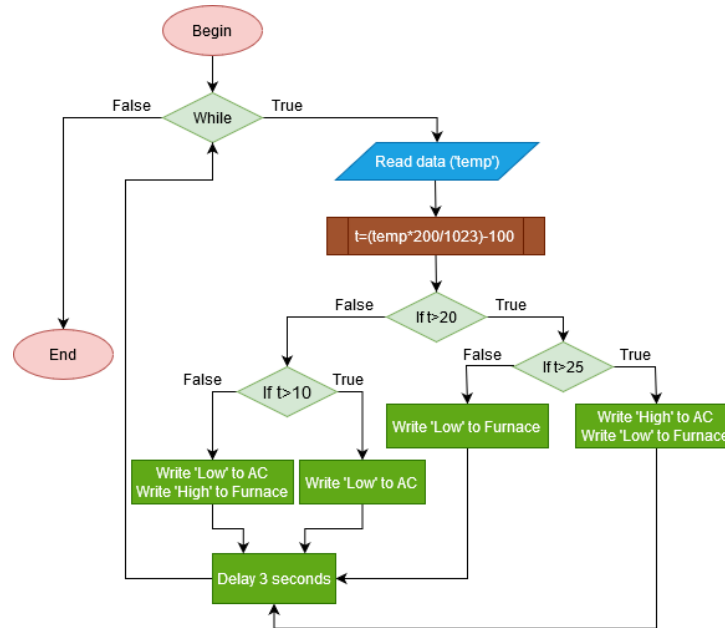


Figure B.1. Room temperature monitoring flowchart.

Figure B.2 illustrates the room temperature monitoring python code:

```
1 from gpio import *
2 from time import *
3
4 def main():
5     pinMode(0, OUT) #AC pin
6     pinMode(1, OUT) #Furnace pin
7     pinMode(7, IN) #Temperature sensor pin
8     print("Getting data...")
9     while True:
10        temp=analogRead(7);
11        t=((temp*200)/1023)-100;
12        if (t>20):
13            if (t>25):
14                digitalWrite(0, HIGH);
15                digitalWrite(1, LOW);
16            else:
17                digitalWrite(1, LOW);
18        else:
19            if (t>10):
20                digitalWrite(0, LOW);
21            else:
22                digitalWrite(0, LOW);
23                digitalWrite(1, HIGH);
24        delay(3000);
```

Figure B.2. Room temperature monitoring python code.

APPENDIX C

Figure C.1 is a flowchart for the kitchen smoke monitoring:

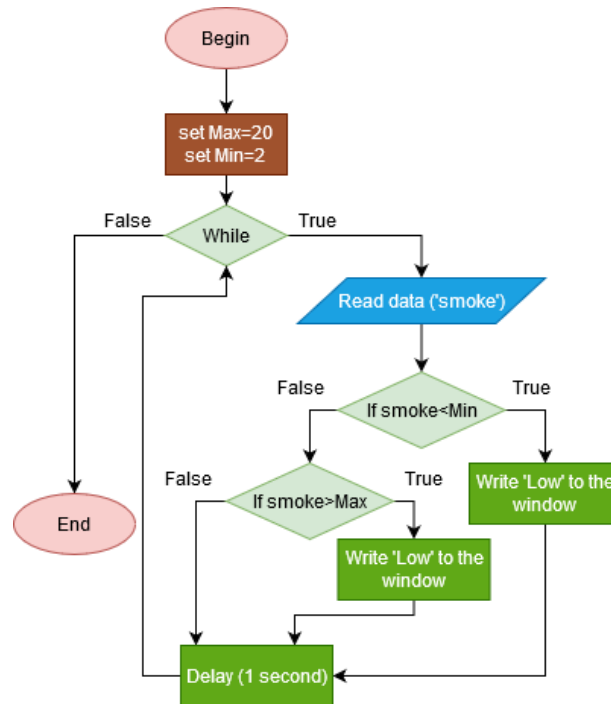


Figure C.1. Kitchen smoke monitoring flowchart.

Figure C.2 illustrates the kitchen smoke monitoring python code:

```
1  from gpio import *
2  from time import *
3
4  def main():
5      pinMode(0, OUT) #Window pin
6      #A0 is the smoke sensor pin
7      MIN=2.0
8      MAX=20.0
9      while True:
10         smoke=float(customRead(A0))
11         if smoke<MIN:
12             customWrite(0,[0])
13         elif smoke>MAX:
14             customWrite(0,[1])
15         sleep(1)
```

Figure C.2. Room temperature monitoring python code.

APPENDIX D

Figure D.1 illustrates the JavaScript code for fire's IR property.

```
1  var state = 0; // 0 off, 1 on
2
3  function setup() {
4      IoEClient.setup({
5          type: "Fire",
6          states: [{
7              name: "On",
8              type: "bool",
9              controllable: true
10         }]
11     });
12 }
13
14 function mouseEvent(pressed, x, y, firstPress) {
15     if (firstPress)
16         setState(state ? 0 : 1);
17 }
18
19 function setState(newState) {
20     state = newState;
21     digitalWrite(1, state ? HIGH : LOW);
22     if ( state === 0 )
23         setDeviceProperty(getName(), 'IR', 0);
24     else
25         setDeviceProperty(getName(), 'IR', 900);
26     setDeviceProperty( getName(), "state", state);
27 }
```

Figure D.1. JavaScript code for fire's IR property.

This component will have a direct-use property (ALT+Click), we made different visual icons to differ between the presence and the absence of the fire. Figure D.2 bellow shows the difference.



Figure D.2. The two states icons for the Fire.

BIBLIOGRAPHY

- [1] Sunshine, Carl. A. (1989). A Brief History of Computer Networking. Computer Network Architectures and Protocols.
- [2] Saravanan Anna malai. INTRODUCTION TO NETWORKING. March 2012 Open University Malaysia.
- [3] Yekini Nureni. DATA COMMUNICATION & NETWORKING. November 2015 Yaba College of Technology.
- [4] Adel Ismail Al-Alawi. Wi-Fi Technology: Future Market Challenges and Opportunities. Journal of Computer Science · January 200. University of Bahrain.
- [5] Tolotriniaina Mirado Rajaonarison. Use Of Petri Net For Studying A Switching Node In A WANnetwork. 2010. fahal-00691520f
- [6] Khouchane K., Rabouhi S. Study of the installation of a fiber network optical. Bejaia 2005.
- [7] Andrew S. Tanenbaum, David J. Wetherall. Computer Networks. 5th edition. Pearson. Janvier 2010.
- [8] M. Yaibuates and R. Chaisricharoen, "ICMP based Malicious Attack Identification Method for DHCP," Chiang Rai, 2014, pp. 1-5
- [9] Han-Chieh Chao, Tin Yu Wn, S. W. Chang and Reen-Cheng Wang, "The network topology-based domain name service," Aizu-Wakamatsu, Japan, 1999, pp. 528-533,
- [10] Rouse, Margaret (2019). "Internet of things (IoT)". IOT Agenda. 14 August 2019.
- [11] Marco Zennaro. Introduction to the Internet of Things. PhD Telecommunications. The Abdus Salam International Centre for Theoretical Physics Trieste, Italy. 2004.
- [12] Rob van Kranenburg. Introduction to the Internet of Things. ARM testimonials (pp.1-10). September 2013
- [13] K Abirami, Anup Chaudhari. Internet of Things (IoT) | Set 2. 2020.
- [16] Stephen McQuerry. CCNA Self-Study: Network Media (The Physical Layer). Cisco Press. Apr 9, 2004.
- [19] Sayed M. and Ali. G. Performance evaluation of a network using simulation tools or packet tracer. IOSR Journal of Computer Engineering (IOSR-JCE), 19:1–5, 2017.

WEBOGRAPHY

[14] Basic networking. Geek University. <https://geek-university.com/ccna>

[15] Computer Hope. Free computer help since 1998. <https://www.computerhope.com/>

[17] Network Encyclopedia. <https://networkencyclopedia.com>. 2020 Copyright.

[18] Cisco Packet Tracer Data Sheet.